AD-A035 166 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MASS F/G 17/2
SPEECH UNDERSTANDING SYSTEMS. VOLUME II. ACOUSTIC FRONT END.(U)
DEC 76 W A WOODS, M BATES, G BROWN, B C BRUCE N00014-75-C-0533
UNCLASSIFIED BBN-3438-VOL-2 NL

END
DATE FILMED
3-77

BOLT BERANEK AND NEWMAN INC

CONSULTING • DEVELOPMENT • RESEARCH

BBN Report No. 3438

12

SPEECH UNDERSTANDING SYSTEMS

Final Report

November 1974 - October 1976

Volume II: Acoustic Front End

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 2904

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-75-C-0533.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: BBN [redacted] 3438 - Vol-2 (14)	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle): SPEECH UNDERSTANDING SYSTEMS. Volume II. Final Technical Progress Report 30 October 1974 to 29 October 1976 Acoustic Front End. (6)	5. TYPE OF REPORT & PERIOD COVERED: Final [redacted] Report. 30 Oct [redacted]74 to 29 Oct [redacted]76 / 6. PERFORMING ORG. REPORT NUMBER: BBN Report No. 3438
7. AUTHOR(s): W. Woods, M. Bates, G. Brown, B. Bruce, C. Cook, J. Klovstad, J. Makhoul, B. Nash-Webber, R. Schwartz, J. Wolf, V. Zue.	8. CONTRACT OR GRANT NUMBER(s): N0014-75-C-0533 (15)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Mass. 02138	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 5D30 (12) 92 p.
11. CONTROLLING OFFICE NAME AND ADDRESS: ONR Department of the Navy Arlington, Virginia 22217	12. REPORT DATE: Dec [redacted]76 (11) / 13. NUMBER OF PAGES: 91
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) (10)

William A. Woods, Madeline Bates, Geoffrey Brown, Bertram C. Bruce, Craig C. Cook

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Acoustic-phonetic experiment facility, acoustic-phonetic recognition, acoustic-phonetic rules, artificial intelligence, ATN grammars, audio-response generation, budget management, computational linguistics, control strategies, data base, dictionary expansion, discourse model, fact retrieval formal command language, formant tracking, grammars, HWIM, knowledge sources

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This five-volume final report is a review of the BBN speech understanding system, covering the last two years of the project from October 1974 through October 1976. The BBN speech understanding project is an effort to develop a continuous speech understanding system which uses syntactic, semantic, and pragmatic support from higher level linguistic knowledge sources to compensate for the inherent acoustic indeterminacies in continuous spoken utterances. These knowledge sources are integrated with

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

060 100

19. Key words (cont'd.)

lexical retrieval, likelihood ratio, linear prediction, multi-component systems, natural language retrieval system, natural language understanding, parametric modeling, parsing, pattern recognition, phonetic labeling, phonetic segmentation, phonological rules, phonology, pragmatic grammar, pragmatics, probabilistic labeling, probabilistic lexical retrieval, prosodics, question-answering, recognition strategies, resource allocation, response generation, scoring philosophy, semantic interpretation, semantic networks, semantics, shortfall algorithm, shortfall density, signal processing, spectral matching, speech, speech analysis, speech generation, speech synthesis, speech recognition, speech understanding, SUR, syntax, synthesis-by-rule, system organization, task model, user model, word verification.

20. Abstract (cont'd.)

sophisticated signal processing and acoustic-phonetic analysis of the input signal, to produce a total system for understanding continuous speech. The system contains components for signal analysis, acoustic parameter extraction, acoustic-phonetic analysis of the signal, phonological expansion of the lexicon, lexical matching and retrieval, syntactic analysis and prediction, semantic analysis and prediction, pragmatic evaluation and prediction, and inferential fact retrieval and question answering, as well as synthesized text or spoken output. Those aspects of the system covered in each volume are:

Volume I.	Introduction and Overview
Volume II.	Acoustic Front End
Volume III.	Lexicon, Lexical Retrieval and Control
Volume IV.	Syntax and Semantics
Volume V.	The Travel Budget Manager's Assistant

SPEECH UNDERSTANDING SYSTEMS

Final Report

November 1974 - October 1976

ARPA Order No. 2904

Program Code No. 5D30

Name of Contractor:
Bolt Beranek and Newman Inc.

Effective Date of Contract:
30 October 1974

Contract Expiration Date:
29 October 1976

Amount of Contract: $1,966,927

Contract No. N00014-75-C-0533

Principal Investigator:
William A. Woods
(617) 491-1850 x361

Scientific Officer:
Marvin Denicoff

Title:
SPEECH UNDERSTANDING SYSTEMS

Editor:
Bonnie Nash-Webber
(617) 491-1850 x227

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 2904

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-75-C-0533.

The BBN Speech Understanding System: Final Report

This is one of five volumes of the BBN Speech Understanding System Final Report. The Table of Contents for the entire set is given below.

Acknowledgements

To the following people who contributed to the development of the BBN Speech Understanding System:

William A. Woods, Principal Investigator
Madeleine Bates
Geoffrey Brown
Bertram C. Bruce
Laura Gould
Craig C. Cook
Gregory Harris
Dennis H. Klatt
John W. Klovstad
John I. Makhoul
Bonnie L. Nash-Webber
Richard M. Schwartz
Jared J. Wolf
Victor W. Zue

To our secretaries - Beverly Tobiason, Kathleen Starr and Angela Beckwith - for the exceptional diligence, competence, and good humor they have shown throughout the assembly of this report.

"Had we but world enough and time..."
Andrew Marvell, To His Cov Mistress

TABLE OF CONTENTS

A. ACOUSTIC FRONT END

1. Initial Signal Processing

This component of the system digitizes the analog speech signal and computes the basic parameters to be used in the Acoustic-Phonetic Recognition (APR) and Verification components.

a. Real-Time Acquisition

The analog speech signal is fed into an analog-to-digital (A/D) converter, which samples the signal into 12-bit samples at one of two constant rates, F=10 or 20 kHz. The sampled signal amplitude is then normalized such that the maximum amplitude in the utterance is 255. The resultant samples are stored on a disk file using 9 bits each, 4 samples per computer word.

As almost all of the parameters used in the APR are computed from the 0-5 kHz part of the signal spectrum (see Sec. B), a 10 kHz sampling rate would have been adequate. We chose, however, to design the system to allow for a 20 kHz sampling rate in case we decided to use information in the 5-10 kHz region of the spectrum for some purpose, such as classification of place of articulation of obstruents.

b. Parameter Computation

All parameters are computed every 10 ms, giving an analysis rate of 100 frames/second. Subsequent timing considerations for phonetic segments and words are based on this basic frame rate. Except for pitch, all parameter computations are based on an analysis window of 20 ms; pitch extraction uses a 50 ms window.

(1) Zero Crossings (ZC)

The first parameter to be computed from the time signal is ZC, the number of zero crossings in a 20 ms interval. The algorithm is independent of the sampling rate; it merely makes sure that the analysis interval is 20 ms. Thus, for 10 kHz sampling, the interval contains 200 speech samples, and for 20 kHz sampling, the interval contains 400 samples.

(2) LP Analysis

The objective here is to perform a linear prediction (LP) analysis on the 0-5 kHz region of the spectrum, with preemphasis. In order to ensure uniformity of analysis, independent of sampling rate, we do not perform preemphasis in the time domain, but rather in the autocorrelation domain. This couples in nicely with the desired LP analysis, as we shall discuss below.

In the autocorrelation method of LP, the predictor coefficients are computed by solving a set of equations [Makhoul 1975a]:

$$\sum_{k=1}^{p} a_k R'(i-k) = -R'(i), \quad 1 \leq i \leq p, \tag{1}$$

where $R'(i)$ is the autocorrelation of the preemphasized signal, in our case; a_k are the predictor coefficients; and p_k is the number of coefficients or poles in the all-pole model (p=13 in our system). $R'(i)$ is obtained from R(i), the autocorrelation of the nonpreemphasized signal, by the following convolution operation:

$$R'(i) = \sum_{k} b(k) R(i-k), \tag{2}$$

where b(k) is the autocorrelation of the impulse response of the all-zero preemphasis filter. In our system, we use a single-zero preemphasis filter $1-z^{-1}$, which in the time domain is equivalent to simple differencing. The autocorrelation of this single zero filter is

$$b(k) = \begin{cases} 2, & k=0 \\ -1, & |k|=1 \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Substituting in (2), we obtain:

$$R'(i) = 2R(i) - R(i-1) - R(i+1), \quad 0 \leq i \leq p. \tag{4}$$

From (4), it is clear that, in order to compute $R'(i)$ for $0 \leq i \leq p$, we need to know R(i) for $0 \leq i \leq p+1$. The remaining issue, then, is how to compute R(i) that corresponds to the 0-5 kHz part of the signal spectrum.

The method we have chosen depends on whether F=10 or 20 kHz. In either case, the signal is first windowed using a 20 ms Hamming window. If

s(n), $0 \le n \le N-1$, is the original signal, and N is the number of samples in a 20 ms interval, then the windowed signal s'(n) is computed from:

$$s'(n) = [0.54-0.46 \cos(2\pi n/N)]s(n), \quad 0 \le n \le N-1. \tag{5}$$

If F=10 kHz, then R(i) is computed directly from s'(n):

$$R(i) = \sum_{n=0}^{N-1-|i|} s'(n)\, s'(n+i), \quad 0 \le i \le p+1. \tag{6}$$

If F=20 kHz, then R(i) is computed from the 0-5 kHz region of the spectrum, as follows. Using the FFT, we compute the signal spectrum P(k):

$$P(k) = \left| \sum_{n=0}^{M-1} s'(n) e^{-j2\pi nk/M} \right|^2, \quad 0 \le k \le M-1, \tag{7}$$

where M=512* and s'(n)=0 for $N \le n \le M-1$. Note that P(k) is even symmetric about M/2, and thus only M/2 spectral values need be computed. Then, we take the lower half of the spectrum (corresponding to 0-5 kHz), make it even symmetric about M/4, and take the inverse Fourier transform to obtain the autocorrelation R(i):

$$R(i) = \frac{1}{M} \sum_{k=0}^{\frac{M}{2}-1} P(k) e^{-j2\pi ik/(M/2)}, \quad 0 \le i \le \frac{M}{2}-1. \tag{8}$$

Equation (8) is then computed using the FFT. (Direct computation of R'(i) using the DFT can also be used since only the first p+2 values are needed.) The resulting computed autocorrelation is equivalent to having lowpass filtered the 20 kHz sampled signal sharply at 5 kHz, downsampled by taking every other sample, and then computed the autocorrelation directly using (6). The use of the above method in conjunction with LP analysis has been termed "selective linear prediction" [Makhoul 1975b].

- - - - - - - - - - - - - - - - -

*Strictly speaking, M should be set to 1024 to get an accurate estimate of the autocorrelation; otherwise, aliasing of the autocorrelation would occur. However, we have found that such aliasing is negligible in the region of interest $0 \le i \le p+1$ (p=13) for a windowed signal.

We now summarize the procedure for obtaining an LP model over the 0-5 kHz region, with preemphasis. We compute the autocorrelation R(i), $0 \le i \le p+1$, via one of the two methods presented above, depending on whether F=10 or 20 kHz. We then compute the autocorrelation R'(i) of the preemphasized signal using (4). The predictor coefficients are then computed from (1) using the following recursive procedure:

$$E_0 = R'(0) \tag{9a}$$

$$K_i = -[R'(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R'(i-j)]/E_{i-1} \tag{9b}$$

$$a_i^{(i)} = K_i$$

$$a_j^{(i)} = a_j^{(i-1)} + K_i\, a_{i-j}^{(i-1)}, \quad 1 \le j \le i-1 \tag{9c}$$

$$E_i = (1-K_i^2)\, E_{i-1} . \tag{9d}$$

Equations 9b-9d are solved recursively for i=1,2,...,p. The final solution is given by

$$a_k = a_k^{(p)}, \quad 1 \le k \le p . \tag{9e}$$

We have chosen to store on disk the reflection coefficients K_i, $1 \le i \le p$, for every frame. The predictor coefficients can then be obtained, if needed, using the recursion in (9c).

The all-pole LP spectrum S(k) can be computed from:

$$S(k) = \frac{E_p}{\left| 1 + \sum_{n=1}^{p} a_n\, e^{-j2\pi nk/L} \right|^2}, \quad 0 \le k \le L-1 , \tag{10}$$

where E_p is the minimum total LP error obtained in the final recursion from (9d), and L is the total number of equispaced frequency points that are computed on the unit circle. In reality, we need compute only L/2 points, which cover the 0-5 kHz region. In our system, we have chosen L=256, and therefore the frequency spacing between spectral values is 39 Hz. S(k) in (10) is computed efficiently by dividing E_p by the spectrum of the sequence

1, a_1, a_2, ... a_p, which is obtained by taking the magnitude square of the FFT of this sequence, with an appropriate number of zeros added to form a total of L points. The all-pole spectrum S(k) is used in the computation of several parameters, as we shall describe below.

(3) Spectral Energy Parameters

We compute the following parameters, which depend on the spectral energy in different frequency bands of the all-pole spectrum S(k):

ROP = Total energy in the 0-5 kHz band of the preemphasized spectrum S(k).

LE = Energy in the 120-440 Hz band of the non-preemphasized version of S(k).

MEP = Energy in the 640-2800 Hz band of S(k).

HEP = Energy in the 3400-5000 Hz band of S(k).

CM75 = The frequency above which 75% of the energy in S(k) is contained.

In particular, we use the following equations to compute the first four parameters:

$$ROP = 10 \log_{10} R'(0) , \tag{11}$$

$$LE = 10 \log_{10} \sum_{k=3}^{11} S(k)*0.5/[1-\cos(\frac{2\pi k}{L})] , \tag{12}$$

$$MEP = 10 \log_{10} \sum_{k=16}^{72} S(k) , \tag{13}$$

$$HEP = 10 \log_{10} \sum_{k=87}^{128} S(k) . \tag{14}$$

The P in ROP, MEP, HEP, means "preemphasis". The only parameter that uses the "deemphasized" spectrum is LE, and the extra term in (12) performs the deemphasis function, which amplifies low frequencies relative to high frequencies using a real pole at z=1.

We note here that the process of preemphasis tends to remove low-frequency noise, which is prevalent in many environments. Since the LP analysis is performed on the preemphasized signal, the LP spectrum is not affected much by low frequency noise. Now, the deemphasis in (12) is

performed on the LP spectrum, and not on the original spectrum. As a result, LE is relatively independent of low frequency noise, or at least less dependent on it than would have been the case had we measured it from the signal spectrum directly.

(4) Formant Extraction

Our formants are extracted from the poles of the all-pole spectrum S(k). The poles are computed by finding the roots of the polynomial

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} . \qquad (15)$$

We have written a special routine that computes these roots efficiently by making good initial estimates of the poles and their bandwidths from spectral peaks, and then using a Newton-Raphson iteration. The resultant poles are then converted to the s-plane by setting each pole $z_k = \exp(s_k/F)$, where $s_k = 2\,(h_k + jf_k)$ is the corresponding pole in the s-plane; f_k is the frequency of the pole and h_k is its half-bandwidth. If a root $z_k = z_{kr} + jz_{ki}$, then:

$$f_k = \frac{F}{2\pi} \arctan \frac{z_{ki}}{z_{kr}}$$

$$h_k = \frac{F}{4\pi} \log(z_{kr}^2 + z_{ki}^2) . \qquad (16)$$

The computed poles are then used as the raw data for the formant extraction routine. The routine uses a minimum of contextual information, enabling it to track formants left to right through an entire utterance. (Though computed everywhere, those formant values extracted within obstruent regions are neglected by the APR and Verifier.) The formant extraction procedure for a single analysis frame follows.

a) a list is made of all poles with frequency between 150 Hz and 3100 Hz and half bandwidth less than 750 Hz. They are sorted in order of increasing frequency. Pole frequencies will be denoted as f(1), f(2),...f(j), and the corresponding half bandwidths as h(1), h(2),...h(j).

b) F1 is chosen to be f(1) if f(1) is below 900 Hz. If f(2) is within 150 Hz of f(1), and less than 950 Hz, and if its half bandwidth, h(2), is much less than h(1), that is,

if $(f(2)-f(1))*4 < (h(1)-h(2))$

F1 is set to f(2) instead. If there are no narrow bandwidth poles below 900 Hz, there is no F1.

c) If there are only two poles remaining, they are assigned to F2 and F3. In this case processing for this frame is finished.

d) If there are more than two remaining poles, they are re-sorted in order of increasing bandwidth. The first two are put in a separate "working" list.

e) A loop to find F2 and then F3 starts here.

f) The lowest frequency pole in the working list is assigned to the current formant.

g) If the formant just set changed by more than 500 Hz from the preceding frame, then the distance between the new value and the preceding value for the adjacent formant is computed. To do this, we let $F_i(n)$ denote the value of the ith formant picked for frame n. The distance will then depend on the sign of the change in the ith formant:

if $F_i(n)-F_i(n-1) > 0$		$F_i(n)-F_i(n-1) < 0$
	then the distance equals	
$F_{i+1}(n-1) - F_i(n)$		$F_i(n) - F_{i-1}(n-1)$

If this distance is too small, that is,

$$\text{if Distance}*4 < |F_i(n)-F_i(n-1)|$$

and there are more poles remaining then the next lowest bandwidth pole is added to the working list and the loop is started again with step (e).

h) If f(i) was just assigned to F3, and f(i+1)-f(i) < 150 Hz, and has much narrower half bandwidth, that is

$$\text{if } (f(i+1)-f(i))*4 < h(i)-h(i+1)$$

then f(i+1) is assigned to F3 instead.

i) The pole just assigned to F2 or F3 is eliminated. Try to find the next formant (i.e. Go to step (f).)

If the average non-zero F0 for the utterance is greater than 150 Hz, then the values used for formant thresholds in steps (a) and (b) (900 and 950 Hz for the highest F1, and 3100 Hz for the highest F3) are determined according to the average F0 [Schwartz, 1971]. F4 for the entire utterance is set to 3100 Hz in order to facilitate the continuity test on F3 in step (g).

After the formants are computed, they are smoothed using a three point median smoothing described by Tukey [1974] and discussed by Rabiner, et al.

[1975]. This smoothing eliminates most of the errors, without eliminating the fine detail of sudden formant transitions. We found that when median smoothing is followed by a 3-point (1/4-1/2-1/4) Hanning window smoothing, too much of the fine structure in the formant transitions is lost, so we do median smoothing only. Although this formant extraction routine is fairly simple, it has resulted in reasonable formant tracks.

(5) Fundamental Frequency Estimation

Fundamental frequency is estimated for each frame using the downsampled center-clipped autocorrelation algorithm described by Gillmann [1975] with only minor changes. F0 estimation for fundamentals on the order of 100 Hz requires a wider window than the 20 ms width used for the other parameters; a 50 ms window is used, extending 10 ms before and 20 ms after the window used for the other parameters.

The speech waveform is digitally lowpass filtered, downsampled to a 2 kHz sample rate, center clipped, and then 50 ms windows are autocorrelated. The autocorrelation function is examined for peaks. A peak of sufficient amplitude must be found for the frame to be considered voiced. If more than one such peak is found, the peak whose lag is closest to a running average of lags of previously found peaks is picked as representing the pitch period. A 3-point interpolation is used to get a finer estimate of the peak location. The "raw" F0 values obtained every 10 ms as described above are then smoothed by a 3 point median smoother, to eliminate isolated voiced or unvoiced points.

(6) Timing

All parameter computations are performed in floating point on our PDP-10 TENEX time-sharing system. The time taken to compute all parameters, including pole-root finding and pitch extraction, is 55 seconds per second of speech.

B. ACOUSTIC-PHONETIC RECOGNITION

Introduction

This section describes the Acoustic-Phonetic Recognition (APR) component of the BBN Speech Understanding System (HWIM). Its main purpose within the system is to produce as good a transcription of an utterance as possible, using various time varying parameters derived from the digital waveform and short-time spectra, (e.g., energy, formant frequencies). This transcription is used by the Lexical Retrieval component (Vol. 3, Sec. D) in determining the sequence of words making up the utterance. Acoustic-phonetic as well as phonological knowledge of English is employed extensively.

This section is divided into four parts. The first part presents the basic approach of the APR. The second part describes the state of this component as of October 12, 1976. The third part reports some performance statistics for the APR component as of the same date. The fourth suggests some long-term and short-term improvements that might be made.

1. APR Approach

Acoustic-Phonetic Recognition (APR) in HWIM consists of three basic tasks: SEGMENTATION, LABELING, and SCORING. SEGMENTATION is deciding where the phoneme boundaries are. LABELING is determining a rough phonetic characterization of each segment produced by the segmentation phase. (Note that the distinction between these two phases is not always clear cut.) SCORING is determining a score for the correspondence of each phoneme possibility for each segment.

1.1 Segment Lattice

One of the most important aspects of the APR is the use of a data structure called a a Segment Lattice, in order to reduce the chance of segmentation errors. Such a lattice provides alternative segmentation paths (i.e., sequences of segments) in those cases where the correct segmentation is unclear. Fig. 1 illustrates alternative paths spanning a given time region.

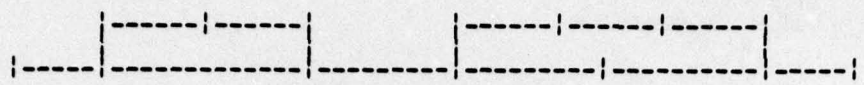


Fig. 1. Segment Lattice.

For further definition and justification of the use of a segment lattice see Schwartz [1975].

1.2 Multiple Pass Strategy

Because the acoustic characteristics of a phoneme vary greatly with its context, it is very helpful to know the nature of that context when making any decision in the construction of a segment lattice. Since context is not available initially, one way to approximate it is to employ a multi-pass APR strategy. Each pass, in general, consists of four steps: initial segmentation, initial labeling, adjustment of segment boundaries, and relabeling. Boundaries are adjusted to correspond to reliable acoustic events. The acoustic events examined in subsequent steps are determined by the results of the initial labeling. Relabeling is then performed using the adjusted boundaries. Each pass operates on regions demarcated by the segmentation in the previous pass, performing more detailed segmentation and labeling by using the more detailed contextual information then available. In this way, acoustic-phonetic rules for segmentation and labeling can be designed for specific phonetic environments.

1.3 Boundary Confidences

While adding optional paths to a segment lattice greatly increases the probability that the "correct" path is represented, it also increases the ambiguity facing the Lexical Retrieval component. In order to partly alleviate this problem, one can include a confidence measure for each boundary in the lattice. If the boundary confidences are then combined with the phoneme-based word match scores (provided they correlate well with the likelihood that the boundary is part of the "correct" path), the word matcher will be better able to choose between the many possible paths through the lattice. In order to compute boundary confidences, a parameter corresponding to the evidence of a boundary is used. For example, the depth of a dip is a good indicator of its reliability as a boundary.

1.4 Experimentation

The parameters, thresholds, and probability density distributions used by the APR program have been determined using the data in a large data base of hand-labeled utterances, with the aid of the Acoustic-Phonetic Experiment Facility (APEF). This provides a highly interactive environment for performing a wide variety of acoustic-phonetic experiments. As the basic facility has already been described in an earlier publication [Schwartz, 1976], it will not be described here. This approach of using an APEF assures that the algorithms developed are realistic and within the capabilities of a computer program.

Since publication of the paper describing this facility, several modules have been incorporated that enable a user to design and test non-parametric multi-dimensional probability distributions that discriminate among a set of phonetic classes. The probability distributions thus derived are then transferred directly to the APR program for use in "selective modification" as discussed in Section B.1.6.

1.5 Probabilistic Labeling

An important factor in designing the APR program was optimizing its interface with Lexical Retrieval via the segment lattice. This was done through the use of probabilities, which though hard to estimate accurately, do afford a well-defined formalism for manipulating and combining scores. In an effort to maintain maximum flexibility in this interface, each segment label consists of an independent score for each possible dictionary phoneme [See Appendix 1]. For our current system, in which 71 different dictionary phonemes are used, there are 71 different scores. Each score represents the probability densities of the relevant acoustic feature values, given that that phoneme is the correct one, divided by the unconditioned probability of those values. That is,

$$\frac{P(\text{Acoustic features values} \mid \text{Phoneme})}{P(\text{Acoustic features values})}$$

This form is used because it is theoretically easy to compute and can be combined with similar scores using Bayes' Rule.

However, computing these probability densities based on all relevant features (of which there is an unbounded set) would be a tremendous task. Therefore, we have created a large set of acoustic labels under which a segment can be classified. [See Appendix 2] These labels are intended to map largely into dictionary phonemes, or classes of phonemes, but they are distinct from the dictionary phoneme set. Using these labels, the segment scoring approximation then becomes:

$$\frac{P(\text{Label} \mid \text{Phoneme})}{P(\text{Label})}$$

These probability ratios are contained in a long-term ph/lab confusion matrix. Thus, in our current system, if the label on a particular segment is Label-j, then there are 71 different phoneme scores for that segment, where each score is:

$$\frac{P(\text{ Label-j} \mid \text{ Phoneme-i })}{P(\text{Label-j})} \qquad \text{for } i = 1 \text{ to } 71$$

we gather these scores by looking at a large amount of speech which has been run through the APR program. The numerator, P(Label-j | Phoneme-i), is equal to the number of times that Label-j was used to label an instance of Phoneme-i (in a one-to-one match), divided by the total number of instances of Phoneme-i. The denominator, P(Label-j), is just the number of times Label-j was used for any phoneme, divided by the total number of instances of any label.

It is desirable to compute the score based on actual observed acoustic feature values, since differences in an acoustic feature that do not cause a change in the segment label are still relevant and can be incorporated in the final score for a phoneme. To do this we employ a technique that we call selective modification.

1.6 Selective Modification

Depending on the original segment label, a small percentage of the dictionary phoneme scores may be re-evaluated (i.e., selectively modified) using particular acoustic feature values extracted from that segment. For instance, the scores on the unvoiced plosive phonemes are computed using:

a) Burst Frequency (parameter CM75 measure at the burst)

b) VOT (the measured time between the burst and the following voiced sound)

c) F3-F2 (measured 2 frames before the silence) - for those unvoiced plosives that appear to be preceded by a sonorant.

d) Energy in burst (ROP measured at the burst).

Using this set of four features, the probability ratio:

$$MOD = \frac{P(\text{ feature set } | \text{ Phoneme-i })}{P(\text{ feature set } | \text{ Phoneme is an unvoiced plosive })}$$

is estimated using a non-parametric probability density estimation technique.

There are three different situations, each requiring a different type of score modification. The simplest case is when the label is very unlikely to be used for the phoneme(s) of interest. For example, if the label is one that indicates a vowel or glide, then all the unvoiced plosives are very unlikely. In this case, it is not worth the computation to make a small adjustment to these scores which are already very low. Therefore, the score on the unvoiced plosive phonemes is unchanged from the confusion likelihood:

$$\frac{P(\text{ Label-j } | \text{ Phoneme-i })}{P(\text{ Label-j })}$$

For those labels that are not unvoiced plosive labels, but are close enough to warrant modification to the unvoiced plosive phonemes (e.g., VPLOS,BV,PB,FTH), the score on each unvoiced plosive phoneme is:

$$\frac{P(\text{ Label-j } | \text{ Phoneme-i })}{P(\text{ Label-j })} * MOD$$

The most modification occurs when the label used by the program indicates some combination of unvoiced plosive phonemes (labels like PK, T, RETPLS). Some of the acoustic features used to decide among the several unvoiced plosive labels are the same acoustic features used to compute MOD. In order to avoid using the same information twice, the scores for the unvoiced plosive phonemes in this third case are:

$$\frac{P(\text{Label=unvoiced plosive} \mid \text{Phoneme=unvoiced plosive})}{P(\text{Label=unvoiced plosive})} * \text{MOD}$$

1.7 Context Dependency

Our multi-pass segmentation and labeling scheme gives us some sense of the phonemic context of each segment in the lattice. However, relying on this apparent context with possibly incorrect hypotheses about the identity of the adjacent segments may lead to labeling errors. In cases where these hypotheses are more likely to be incorrect, it is advantageous to consider all possible relevant contexts, and compute different results for each postulated context.

For example, two of the features used to distinguish among the unvoiced plosives [P,T,K] are burst frequency and voice-onset-time (VOT). However, when an unvoiced plosive is followed by [R], then burst frequency and VOT are considerably different from the case where it is followed by a vowel. Since some of the [R] transition is often unvoiced when it follows an unvoiced plosive, it is not always possible to determine absolutely whether the plosive is followed by a vowel or by [R]. Therefore, we might consider two (or more) allophones of each plosive; one followed by vowels, the other followed by [R]. For instance, the score on [T R] would be the probability that the relevant acoustic parameters have their particular values, given that the phoneme this segment represents is a [T] and that it is followed by [R].

When used in word matching, only the score of the appropriate allophone of [T] need be examined. These allophonic variations can be compiled into the dictionary using the same procedure developed for application of generative phonological rules [See Vol. III, Sec. B]. Of course, one wants to minimize the number of different allophones that need to be considered, but a reasonable balance can result in a large improvement in word matching. In the remainder of this section the terms "dictionary phoneme" and "allophone" will be used interchangeably.

1.8 Speaker Normalization

In developing the APR component, we have not incorporated decision thresholds based on speech characteristics measured on each individual speaker. We have instead based the APR decision logic on a data base of utterances by several speakers. Specific measurements are generally adaptive to the utterance being processed. Formants are normalized according to the average fundamental frequency for the utterance [Schwartz, 1971], and most thresholds on energy are relative to minimum or maximum values measured from the utterance itself. We feel that by taking this approach, we might be more likely to find those acoustic cues which are applicable to the speech of many speakers.

The data base from which the APR was developed consists primarily of sentences spoken by five male speakers. The table below indicates the linguistic background of the speakers.

Speaker	Age	Height	Geographical Origin
DHK	38	6'2"	Whitefish Bay, Wisconsin
DWD	33	6'4"	Suburban St. Louis, Mo.
JJW	34	6'0"	Suburban Philadelphia
RMS	26	5'10"	New York City
WAW	34	5'10"	Charleston, W.Va.

2. Acoustic-Phonetic Recognition Program

HWIM's current APR component embodies most aspects of the approach described above.

2.1 Acoustic Parameters

Table 1 gives the parameters currently used by the program. The "Z" in the first three parameter names indicates that the parameter has been smoothed by a 3-point (1/4-1/2-1/4) zero-phase filter. For a more detailed definition of these parameters, see Section A.

Name	Definition	Use of Parameter
LEZ	Smoothed energy in the region from 120-440 Hz.	Sonorant obstruent segmentation, aid in voicing decision on fricatives
MEPZ	Smoothed energy in the preemphasized spectrum from 640-2800 Hz.	Segmentation of non-vowels within sonorant regions.
HEPZ	Smoothed energy in the preemphasized spectrum from 3400-5000 Hz.	Segmentation of non-vowels within sonorant regions. Unvoiced-plosive detection.
ROP	Energy in the preemphasized spectrum	Burst location, plosive and fricative identification, and many others.
F1 F2	Formant frequencies	Detecting glides and nasals within sonorant regions, segmenting vowel regions, labeling vowels and glides. Formant transitions are used for labeling consonants.
F0	Fundamental frequency	Normalizing formants, aid in voicing decision.
CM75	"75% center of mass" indicates rough spectral shape.	Used in identifying fricatives and unvoiced plosives.
ZC	Zero crossings	Helps in detecting the end of the utterance.

Table 1. Parameters used by the APR.

2.2 APR Procedure

The program begins by applying a general dip detector to each of the three wide band energy parameters. Then, the whole utterance is segmented into sonorant regions and obstruent regions, based on the dips found in the low frequency band. Dips found in the middle and high frequencies are taken as an indication of possible nasals, glides, or voiced obstruents (e.g., V,DH,HH,DX). Within regions initially classified as obstruent, the dips in the high frequency band are used to segment strident fricatives from plosives and weak fricatives. The plot in Fig. 2 shows the result of this preliminary segmentation for the utterance "I will fly to San Diego". The horizontal axis represents time in seconds. Directly above the time axis is a manual transcription of the utterance, and above that are the three wide band energy parameters. Superimposed on each is the result of the dip detector. At the top is shown the first stage segmentation.

This preliminary segment lattice has no branching and only distinguishes among a few broad categories. Some of the regions generated by this initial phase contain more than one phoneme, so within each region, parameters specific to the type of region are used for further segmentation and labeling.

Following this first phase, an ordered set of 35 region-specific acoustic-phonetic rules is applied to the lattice. The main body of the APR procedure consists of these rules, which delete branches, add branches, and change or narrow the label on any segment. While these rules are not defined in any rule formalism, being arbitrary BCPL programs, they are thought of as such because of their method of application. Each rule depends on previous rules having been applied beforehand and applied over the entire utterance wherever appropriate. Two examples of simple rules are given below.

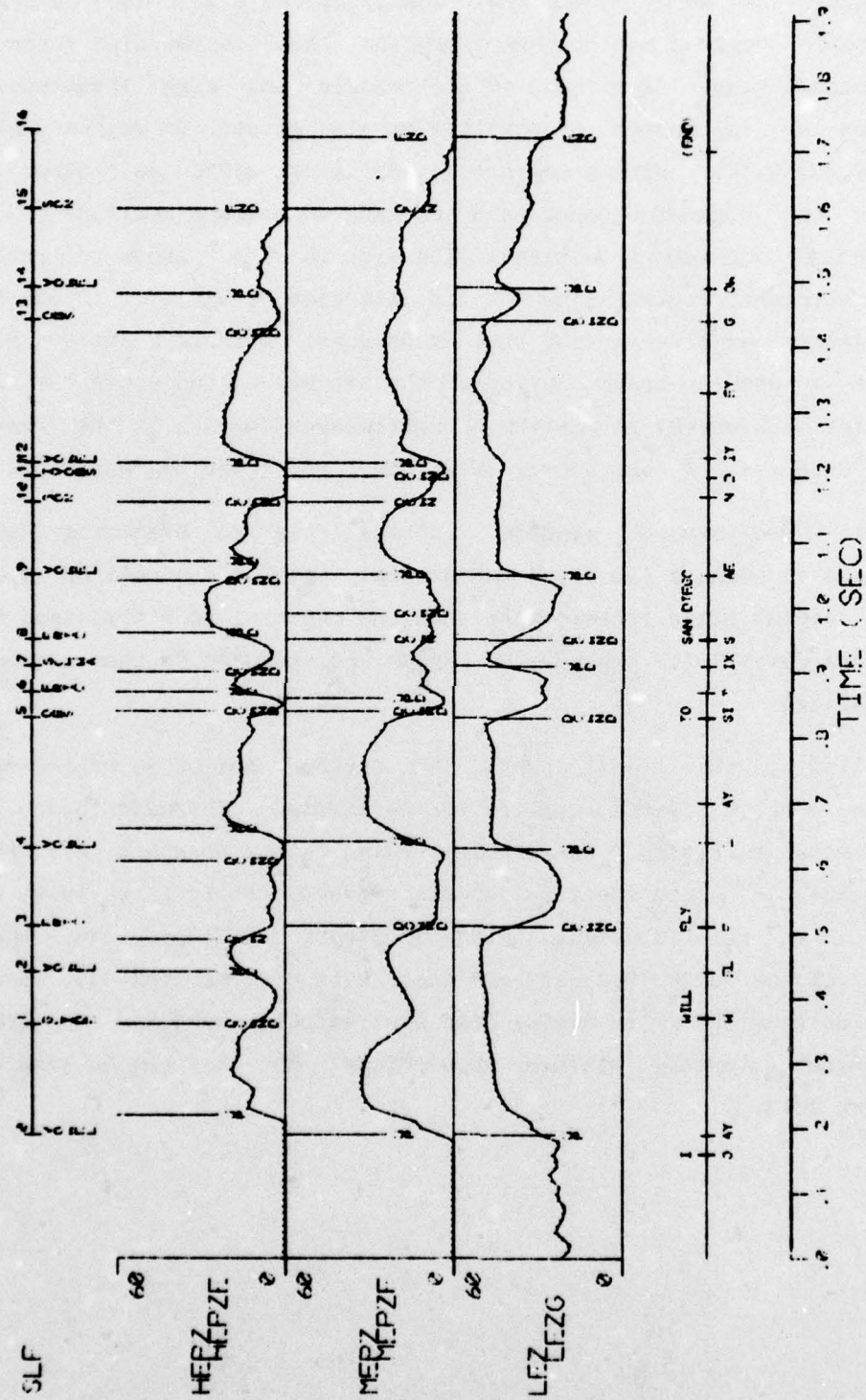


Fig. 2 Preliminary Segment Lattice

Rule 7d: lipsmack

Sometimes, at the beginning of an utterance, when the speaker opens his mouth to begin talking, there is a short noise or click, usually due to the tongue or lips. This comes out as a short high frequency period followed by silence. Since an English sentence cannot start with phonemes that would look like a short fricative followed by a silence, this can be easily detected and eliminated.

```
// Check for Lip Smack at beginning, defined as a short
// FRIC followed by a long OBS

SortBdrys()
let AfterLipSmack := 0
{Lipsmack
      // See if the utterance starts with a FRIC
 let s := FindSeg(1,2,FRIC)
 if s>0 then
      // See if it's followed by an OBS
 { let so := FindSeg(2,0,OBS)
   if so>0 then
      // Check duration of FRIC
   { let lbt,rbt := lhz bdrys|1, lhz bdrys|2
     if rbt-lbt < 400 then
     { PWriteF(" Lip Smack at 'i' in 's/",lbt,UTNAM)
       DelSeg(so); if s>so then s := s - 1
       DelSeg(s); SortBdrys()
     // So can know to start other searches after
     // this point.
       AfterLipSmack := rbt/100+1
     }
   }
 }
 }Lipsmack
```

Fig. 3. Lipsmack rule.

Rule 8d: vplos

Segments labeled as PLOS are checked for a burst towards the end. If one is found, and the voice onset time (VOT) is more than 20 msec, then an optional unvoiced plosive is added. If the VOT is small, and there does not appear to be any frication on either side which would shorten the VOT of an unvoiced plosive, then the label is changed to a voiced plosive label.

```
      // Look for a PLOS segment
for s := 1 to nsegs do if q1 segments|s=PLOS then
{vplos
  let lb,rb := q4z segments|s, q3z segments|s
      // Find left and right bdry times
  let lbt,rbt := (lhz bdrys|lb)/100, (lhz bdrys|rb)/100

// Find the burst if there is one.
      // minimum energy during the stop
  let tminr0d := Tminv(R0P,lbt,rbt)
  let thresh := R0P|tminr0d %+ 7.0
      // find first frame at least 7 dB above the minimum
  let abovet := Next(R0P,tminr0d,rbt+1,gtr,thresh)
      // if there is none, then use the point of maximum change
  if abovet<0 then abovet := Tmaxv(DR0P,tminr0d,rbt+1)
      // the first negative second derivative of R0P
  let burst := Next(DDR0P,abovet,rbt+1,less,0,nabsv,uselast) -1

  let VOT := rbt-burst
  if VOT>2 then
  { if burst-lbt<2 then loop
      // define a new bdry.
    let uvb := DefineBdry(burst*100,100)
      // Define optional UVPLOS as 2 parts.
      // They will be merged later.
   DefineSeg(lb,uvb,SI,R0P); DefineSeg(uvb,rb,UVPLOS,R0P)
   loop
  }
      // if there is no FRIC or STFRIC on either side,
      // AND NO UVPLOS OR RETPLS ON LEFT, then is
      // most likely that is a VPLOS
  if (FindSeg(0,lb,FRIC,STFRIC,UVPLOS,RETPLS)=0)&
     (FindSeg(rb,0,FRIC,STFRIC)=0) then
      // change segment label
   q1 segments|s := VPLOS
}vplos
```

Fig. 4. Vplos rule.

Several APR rules deal with specific problems such as detection of sentence initial [HH] or glottal stops, and sentence final unreleased plosives, while other rules deal with more general phenomena. It is of some interest that the ordering of the rules has so far been relatively straightforward. A complete list of the rules and a description of the APR procedure for applying them is given in Appendix 3.

The APR program utilizes a large body of acoustic-phonetic and phonological knowledge in order to detect a wide variety of phonemic events. The APR program attempts to determine both manner and place of articulation for all phonemes. In addition the program detects affricates, flapped dentals, and intervocalic glottal stops. It also detects and labels prevocalic and postvocalic glides, sentence initial [HH] or glottal stops, and unreleased plosives at the end of a sentence. Formant transitions are used in labeling consonants, and duration is used extensively as a cue in all phases of both segmentation and labeling. The program also detects unreleased plosive-plosive pairs, medial pauses, syllabic nasals, and vowel-schwa pairs (such as IY-AX in "give me a list").

2.3 Scoring

After the segmentation and general labeling has been completed, the program enters a phase of probabilistic scoring. A vector of the long-term phoneme/label confusion matrix is used to supply an initial score for every phoneme, based on the segment label. Then, depending on the initial segment label, several acoustic feature values are computed, and scores on some phonemes are modified (selective modification). The probability density functions used for selective modification are derived in the Acoustic-Phonetic Experiment Facility [Schwartz, 1976] and transfered directly to the APR program.

The technique of selective modification was only recently incorporated into the APR program and has only been applied to the nasals, fricatives, unvoiced plosives, affricates, and some of the front vowels. A description of these classes and the parameters used is given in Appendix 4. The computed acoustic features are used to modify the scores on selected dictionary allophones as described in Section B.1.6.

2.4 Path Probabilities

In recent months, we have developed a procedure for scoring different paths in the lattice relative to each other. At the time a segmentation change is made, the different path probabilities must be stored with the first different segment in each newly created alternative. These path probabilities are then added into the scores for that segment.

Since the acoustic evidence for the beginning and end of a sentence is not always sufficient (e.g., Sentence initial [DH] or sentence final unreleased plosives) the APR program provides an ending score for each boundary near the ends of the utterance. This score gives the probability that that boundary represents the end of the utterance. This score is used by Control component to score sentence hypotheses which end at that boundary.

2.5 A Sample Segment Lattice

Figure 5 shows a plot of the segment lattice for a token of the sentence "I will fly to San Diego", with energy and formants plotted below. Segment labels are shown at the beginning of the corresponding boundary marker. It is important to remember that although a segment label may appear to be an ARPABET symbol, it actually represents a vector in our long-term confusion matrix which is subsequently modified.

2.6 APR Speed

The APR component currently requires approximately three times real time to generate a segment lattice from the acoustic parameters for an utterance.

3. APR Performance

The performance measures given below are intended mainly for comparing successive versions of this program. That is, even though analogous measurements can be made on the acoustic components of other systems, they are not always strictly comparable. For example, while the use of a segment lattice will decrease the number of segmentation errors, the ambiguity and complexity it adds may outweigh the good effects. Labeling

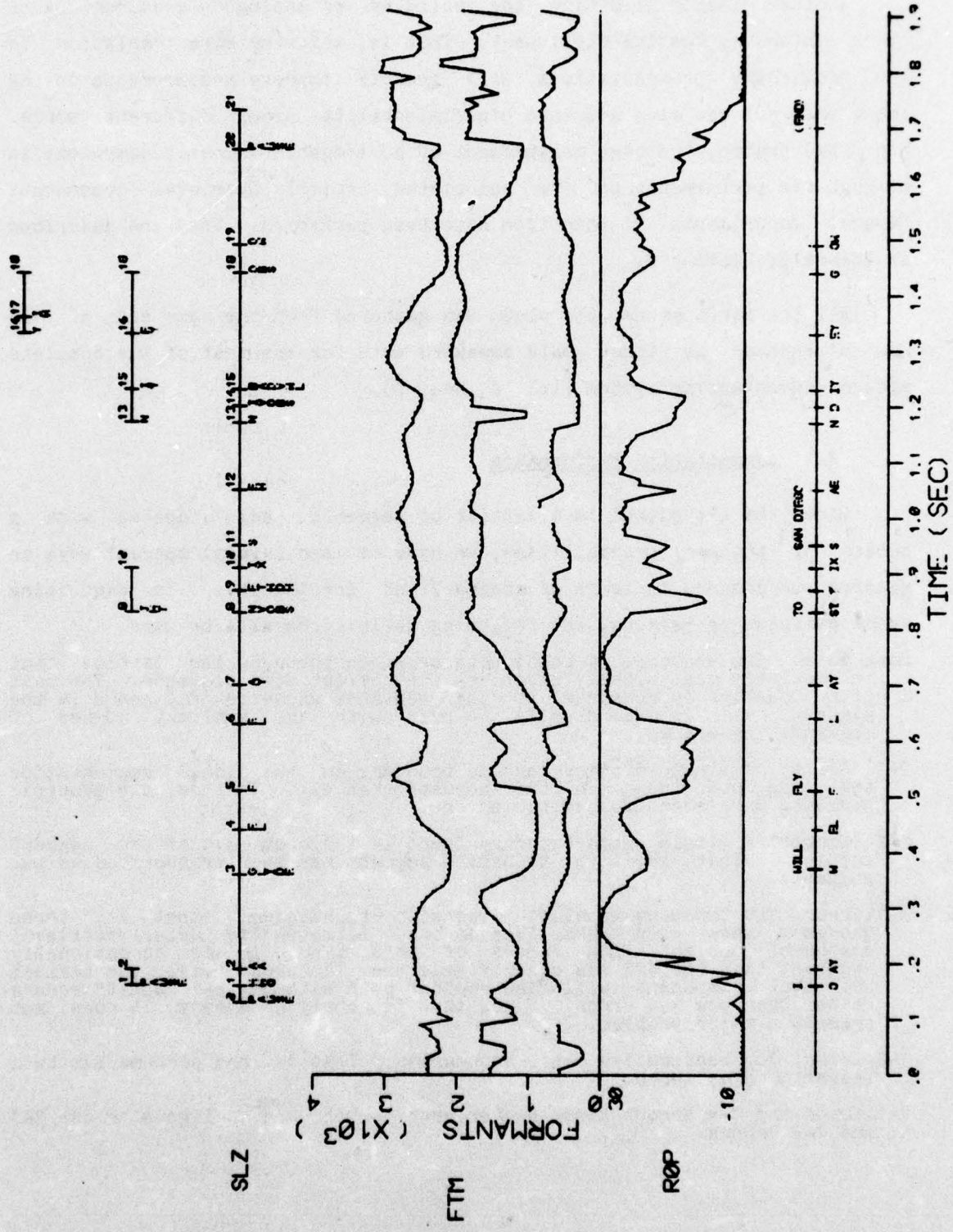


Fig. 5 Sample Segment Lattice

performance is also likely to appear better with a lattice, since it is only measured on the best path.

Another factor limiting the validity of analogous measurements on other systems is the the dictionary. That is, allowing more variation in the dictionary pronunciations will greatly improve measurements on the front end, but may also decrease discriminability among different words. For this reason, the best measurement of an Acoustic-Phonetic component is through the performance of its associated Lexical Retrieval component. Several experiments of this type have been performed. They are described in Volume 3, Section D.

All the performance data shown was gathered from the same set of 124 new utterances by three male speakers used for the test of the complete speech understanding system [Vol. I, Sec. D].

3.1 Segmentation Performance

Since the APR output is a lattice of segments, each labeled with a vector of phoneme probabilities, we have devised several special ways to measure performance in terms of accuracy and specificity. In describing these performance measure, the following definitions will be used.

Best Path: The sequence of contiguous segments through the lattice that subjectively most closely underlies the correct word sequence. The best path usually corresponds to the sequence whose lexical score is the highest. This is also usually the path with the smallest number of segmentation errors.

2-1 "Split": A single missing phoneme boundary in the ideal segmentation that was not found in the acoustics at all. That is, two phonetic segments have been transcribed as one.

1-2 "Merge": A single extra boundary found in the best path of the segment lattice. That is, one phonetic segment has been transcribed as two segments.

3-1 Error: Two consecutive missing phonetic boundaries. That is, three phonemes have been transcribed as one. Although the Lexical Retrieval component does not allow errors of this type, it is occassionally apparent that the APR has clearly made one. In such a case, the Lexical Retrieval component will find another path with two 2-1 "Split" errors rather than one 3-1 error. Since this is rarely necessary, it does not present a major problem.

1-3 Error: Two consecutive extra boundaries. That is, one phoneme has been transcribed as three.

4-1 Error and 1-4 Error: These higher order errors are analogous to the 3-1 and 1-3 errors.

Branching Ratio: The total number of segments in the lattice divided by the number of boundaries (excluding the last boundary), i.e., the average number of different segments emanating to the right from any boundary.

Segment expansion ratio: The total number of segments in the segment lattice divided by the number of phonemes in the utterance.

Average depth of the lattice: Average number of paths crossing any point. It is sampled immediately following each non-final boundary in the lattice.

Table 2 describes the segmentation performance of the APR program.

Total phonemes in ideal segmentation	2850	
Total boundaries in lattices	3940	
Total segments in lattices	5127	
Branching Ratio (segments/boundaries)	1.35	
Segment Expansion Ratio (segments/phonemes)	1.79	
Average Lattice Depth	1.85	
Total 2-1 "Splits"	46	1.6%
Total 1-2 "Merges"	48	1.8%
Total 3-1 "Errors"	1	0.035%
Total 1-3 "Errors"	8	0.28%
Total 4-1 "Errors"	0	0
Total 1-4 "Errors"	1	0.035%

Table 2: Segmentation performance

The three measures of lattice ambiguity - branching ratio, segment expansion ratio, and average lattice depth - taken together, indicate the ambiguity added due to the multiple paths in the segment lattice. Note that taking the average branching ratio alone can be misleading, as can be seen in Fig. 6 in which the two lattice sections have equal average branching ratios.

b.r. = 3/2 = 1.5 b.r. = 6/4 = 1.5

(a) (b)

Figure 6

Clearly, the ambiguity in section (b) is higher than that in section (a), evidenced by their different segment expansion ratios which indicate whether most of the optional segments are short segments added to avoid missing boundaries ("splits") or long segments bridging two or more correct

segments added to avoid extra boundaries ("merges"). Depending on whether the lattice sections in Fig. 6 represent one phoneme or two, the segment expansion ratio in (a) would be 3 or 1.5 respectively, while in (b), it would be 6 or 3.

The average depth is also an incomplete measure of lattice ambiguity. For instance, in the two lattice sections shown below, both the branching ratio and the segment expansion ratio are identical, while the average depth is greater for the example on the right. Since both sections contain the same number of distinct paths, one would like to say that the ambiguity due to the lattice structure is the same for both.

avg. depth = 1.67 avg. depth = 2.0

3.2 Labeling Performance

Since a segment is labeled with a score (probability ratio) for each of the 71 possible dictionary symbols, it is not appropriate to measure "labeling errors" as such. However, we can measure label selectivity in several ways. One possible measure is the percentage of times that the correct phoneme scores highest or, more generally, the likelihood that the correct phoneme is within the top N choices. This performance measure is highly dependent on the number of allophones defined in the system. The following table shows the percentage of the time that the correct phoneme is within the top N choices. For example, 69% of all the segments were labeled correctly within the top 2 choices.

N	1	2	3	4	5	6	7
	52%	69%	75%	80%	83%	85%	86%

However this measure does not adequately show label selectivity. That is, if the correct phoneme on a segment has the highest score, one also wants to know how far below it the second highest score falls. On the other hand, if the correct phoneme is not first, then the difference between its score and that of the top (incorrect) phoneme is important in addition to its rank.

We measure these differences as the log of the ratio of the two probabilities in question. (In the following, a correct segment is a segment in the best path whose highest phoneme is correct. An incorrect segment is a segment in the best path whose highest scoring phoneme is incorrect.) For each correct segment, we record the log of the ratio of the two highest phoneme scores. For each incorrect segment, we record the log of the ratio of the highest score to that of the correct phoneme. The log ratios are summed separately for correct and incorrect segments.

A measure of label specificity that is independent of labeling accuracy is a comparison of the average log ratios for correct and incorrect segments. In the definition of these two numbers below, P_i is the probability ratio that ranks i among the 71 allophone scores.

$$\frac{\sum^{N_{cor}} \text{Log}_{10}\left(\frac{P_1}{P_2}\right)}{N_{cor}} \quad \text{vs} \quad \frac{\sum^{N_{inc}} \text{Log}_{10}\left(\frac{P_1}{P_{cor}}\right)}{N_{inc}}$$

Normally, the average for incorrect segments will be larger than that for correct ones, since when the first choice is incorrect, the correct choice is not restricted to the second choice, while when the first choice is correct, the comparison is always with the second choice. For correct segments in the test set of 124 new utterances, this average log ratio is 0.58. For incorrect segments it is 0.78. The ratio of the sum for correct segments to the sum for incorrect segments can be regarded as a measure of labeling accuracy and specificity. In the development of our scoring algorithms we have attempted to maximize this measure.

$$\frac{\sum^{N_{cor}} \text{Log}_{10}\left(\frac{P_1}{P_2}\right)}{\sum^{N_{inc}} \text{Log}_{10}\left(\frac{P_1}{P_{cor}}\right)}$$

Its rise over the past year has been taken to indicate improvement over previous versions of the APR program.

3.3 Subclass Labeling Performance

The following is a description of the APR's ability to discriminate among the phonemes within a given class, for instance the unvoiced plosive allophones. This will be given only for those classes whose phonemes have undergone selective modification (see Sec. B.2.2).

3.3.1 Discriminating among Unvoiced Plosives

The eight unvoiced plosive allophones that undergo selective modification are [P,KA,K,TG,TV,ST,CH,JH]. [KA] is a /K/ which is followed by a back vowel or a glide within the same syllable. [TG] is a /T/ followed by a glide within the same syllable. [ST] is a prevocalic /T/ preceded by a strident fricative. [TV] is prevocalic /T/ not preceded by a strident fricative. (The principal acoustic difference between the two preceding allophones is the VOT.) The affricates [CH,JH] are treated as unvoiced plosives, since the type of measurements made are similar. The confusion matrix shown in Fig. 7 indicates the unvoiced plosive allophone that had the highest score.

		Correct Allophone							
		P	KA	K	TG	TV	ST	CH	JH
Highest Scoring Phoneme	P	46	12	6	1	10	0	1	2
	KA	5	16	2	1	3	0	0	0
	K	4	1	26	1	0	0	0	0
	TG	1	3	4	47	2	0	1	0
	TV	0	1	3	7	35	1	0	10
	ST	4	0	6	2	8	25	0	4
	CH	1	1	1	2	3	0	7	4
	JH	11	0	0	1	6	0	3	23

Fig. 7: Confusion Matrix for Unvoiced Plosives

The following table indicates the percentage of the time that the correct choice fell within the top N of 8 choices.

N	1	2	3	4	5	6	7
	62%	79%	85%	89%	92%	94%	97%

The average log ratios and the ratio of the sums of log ratios on correct segments to incorrect segments is given below:

Avg. Log ratio for correct segments	1.16
Avg. Log ratio for incorrect segments	1.10
Ratio of correct sum to incorrect sum	1.70

Although we would generally expect that the average log ratio for incorrect segments is greater than that for correct segments, the two averages are approximately equal. This indicates that the phoneme scores are closer together when the top choice is incorrect.

3.3.2 Discriminating among Fricatives

The 8 fricative allophones are [F,TH,V,DH,S,Z,SH,ZH]. The confusion matrix of highest scoring choices is shown in Fig. 8.

Highest Scoring Phoneme \ Correct Allophone	F	TH	V	DH	S	Z	SH	ZH
F	29	2	3	2	3	0	0	0
TH	6	3	1	2	13	2	2	0
V	5	0	19	15	0	7	0	0
DH	6	1	10	32	0	3	0	0
S	0	0	0	0	135	17	1	0
Z	0	0	0	0	43	35	3	0
SH	0	0	0	0	3	0	20	0
ZH	3	0	1	2	0	0	0	0

Fig. 8: Confusion Matrix for Fricatives.

Percent within N choices out of 8:

N	1	2	3	4	5	6	7
	64%	88%	92%	94%	97%	98%	100%

Avg. Log ratio for correct segments	0.64
Avg. Log ratio for incorrect segments	0.72
Ratio of correct sum to incorrect sum	1.55

The ratio measure was 1.15 for the same eight fricative allophones before selective modification, indicating that there was a net improvement.

3.3.3 Discriminating among Nasals

The 5 nasal allophones are [M,N,YM,YN,NX]. The allophones [YM,YN] are instances of phonemes /M,N/ that are preceded by any of [IY,EY,ER,AXR,R]. The same measures are shown below.

Highest Scoring Phoneme \ Correct Allophone	M	N	YM	YN	NX
M	23	4	0	0	1
N	7	93	0	5	1
YM	16	3	6	0	2
YN	1	20	5	10	6
NX	0	1	0	1	5

Fig. 9: Confusion Matrix for Nasal.s

Percent correct within N choices out of 5.

N	1	2	3	4
	69%	86%	92%	98%

Avg. Log ratio for correct segments 0.89
Avg. Log ratio for incorrect segments 0.46

Ratio of correct sum to incorrect sum 3.62

Note that when discriminating among the nasal allophones, when the first choice was correct, the program was much more sure of the decision than when it was incorrect. The ratio measure was 1.09 for the same five nasal allophones before selective modification. In this case, there was a very large improvement.

As described earlier in this section, the reason for introducing the allophones [YM,YN] was that these were sometimes mislabeled as [NX], while other /M,N/ phonemes were not as easily confused. As can be seen from the confusion matrix above, [M] and [N] are often called [YM] and [YN] on the first choice. In order to illustrate that the [M,N,NX] decision is better, the results below show the discrimination among these three allophones:

Highest Scoring Phoneme	Correct Allophone: M	N	NX
M	34	5	2
N	9	109	3
NX	4	7	10

Fig. 10: Confusion Matrix for [M,N,NX]

Percent within N choices out of 3

N	1	2
	83%	97%

Avg. Log ratio for correct segments 1.07
Avg. Log ratio for incorrect segments 0.33

Ratio of correct sum to incorrect sum 16.62

The comparison of the two average log ratios, and the ratio of sums measure are highly dependent upon the number of choices. Therefore, these results are not comparable between different size sets.

3.3.4 Discriminating among Front Vowels

The 5 front vowel allophones that undergo selective modification are [IY,IH,EH,EY,AE]. The same measures are shown below.

Highest Scoring Phoneme	Correct Allophone: IY	EY	IH	EH	AE
IY	89	12	14	4	0
EY	3	40	0	4	0
IH	6	1	82	20	4
EH	1	4	13	68	16
AE	1	2	2	10	31

Fig. 11: Confusion Matrix for Front Vowels

Percent within N choices out of 5

N	1	2	3	4
	72%	90%	96%	98%

Avg. Log ratio for correct segments	1.24
Avg. Log ratio for incorrect segments	0.76
Ratio of correct sum to incorrect sum	4.31

The ratio measure was 1.51 for these five front vowel allophones before selective modification.

In conclusion, the selective modification improved the label selectivity considerably for those classes of allophones in which the technique was applied.

4. Possible Improvements

There are several changes that would improve the performance of the APR program. These improvements fall into three main categories: generalization of the techniques currently used, addition of new information and techniques, and elimination of inconsistencies in some of the methods currently used. Several of the improvements described below have already been implemented in a version of the APR program more recent than that used in the final (12 October) version of HWIM, in which case they will be so cited.

4.1 Generalization of Techniques

The most recent capability to be added to the APR program is selective modification. As described earlier, we have only implemented it on the fricatives, unvoiced plosives and affricates, nasals, and five of the front vowels. Discrimination among members of these classes, which constitute approximately 40% of all phoneme occurrences has improved substantially. Additional improvement can be expected from implementing selective modification for the remaining classes.

There are several labeling decisions in the early stages of the APR that are binary decisions based on a set of thresholds (see Appendix 3). These decisions were implemented in this manner because in the set of approximately 70 utterances initially used in designing the APR, these thresholds were found to yield correct decisions nearly all the time. However, when we expanded the data base, we found that they are not always

correct. One of the disturbing characteristics of statistical scores based on discrete events is that as binary decisions are made more accurate, the few errors that remain are worse. For example, if a decision between nasals and glides is correct 90% of the time, with 10% of the glides incorrectly labeled as nasals, the score for the glides on those incorrectly labeled segments will be proportional to that 10% error rate. If the correct decision rate is improved to 95%, then while the number of incorrectly labeled glides will be halved, their scores will be proportional to the 5% error rate, making them much worse.

This is precisely one of the situations in which selective modification will help, because the score on each phoneme will depend on the particular value of the acoustic feature rather than on a thresholded value. For those decisions that cause problems, selective modification across broad acoustic classes should be combined with selective modification of phoneme scores within those classes.

4.2 Additions

As described in section B.1.1, if there is a possibility that the initial segmentation of a region is incorrect, then the program produces alternate segmentations. Currently, the different competing paths in a segment lattice are all considered equally probable, which is clearly not correct. The relative likelihood of each path should depend on the particular value of the parameter that indicated the possible error. If this is done properly, adding more segmentation paths does not increase the ambiguity facing the Lexical Retrieval component, although it does require more work to look at all the different segments.

Given two possible paths (A and B) through a region, the relative likelihood of path A would be

$$\frac{P(\text{Acoustic evidence} \mid \text{Segmentation A is correct})}{P(\text{Acoustic evidence} \mid \text{given there is a choice between A \& B})}$$

For example, if evidence for a prevocalic [R] is found at the beginning of a vowel sequence, then an optional path is created with the [R] followed by another vowel sequence.

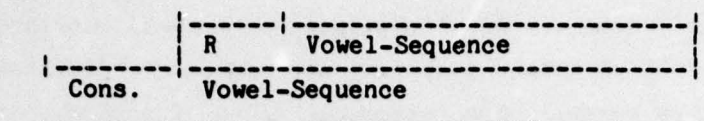


Fig 12: Addition of Prevocalic [R] to the lattice

If the evidence for this comprised both the minimum value of F3 and its rise over the first 5 frames of the region, then the path score on the path with the [R] is

$$\frac{P(\text{min. F3, rise in F3} \mid \text{[R] is correct})}{P(\text{min. F3, rise in F3} \mid \text{given the optional path was created})}$$

while that on the other path is

$$\frac{P(\text{min. F3, rise in F3} \mid \text{[R] is incorrect})}{P(\text{min. F3, rise in F3} \mid \text{given the optional path was created})}$$

Each score would be added to the first divergent segment on its associated path. This sort of score can be produced for most segmentation decisions.

We believe that our discrimination among the fricatives could also be improved. Since voicing in voiced fricatives often drops out part way through, we have tried measuring the parameter LEZ at several different points during the fricative. We found that the best point to measure it is at the boundary between the preceding phoneme and the fricative, rather than around the loudest region. On implementing this minor change, we found that frication discrimination improved as shown below.

		Correct Allophone							
		F	TH	V	DH	S	Z	SH	ZH
Highest Scoring Phoneme	F	26	2	4	2	2	0	6	0
	TH	7	3	1	2	9	2	0	0
	V	8	0	23	16	1	8	0	0
	DH	5	1	5	31	1	0	0	0
	S	0	0	0	0	153	20	0	0
	Z	0	0	0	0	29	34	3	0
	SH	0	0	0	0	2	0	17	0
	ZH	3	0	1	2	0	0	0	0

Fig. 13: Confusion Matrix for Fricatives using LEZ measured at beginning.

Percent within N choices out of 8:

N	1	2	3	4	5	6	7
	67%	86%	92%	93%	96%	97%	99%

Avg. Log ratio for correct segments 0.74
Avg. Log ratio for incorrect segments 0.77
Ratio of correct sum to incorrect sum 1.94

We also found that the addition of simple formant transition measurements (not used here) yields additional improvements in discrimination between the labial and the alveolar and dental fricatives.

We have also considered improving HWIM'S performance through "tuning" its dictionary pronunciations. There are several degrees to which this could be done. If a word in the lexicon has several pronunciations, one could weight their relative likelihoods based on the number of observed occurrences of each pronunciation. Further, one could add a new pronunciation if it scores better than any of the given "legal" ones. For instance, if the [EH] in the word "ten" is usually recognized as [IH] due to the coarticulation effect of the adjacent consonants, then one might include a pronunciation with [IH] in the lexicon, even though the correct phoneme is [EH].

A more drastic step would be to change lexical pronunciations to reflect the characteristics of a particular acoustic component. For instance, if an acoustic component is not designed to recognize diphthongs as single phonetic units, then one might substitute all occurrences of the diphthong [EY] with three new steady-state dictionary symbols (e.g., "EY-left", "EY-middle", "EY-right").

It is clear that these several methods of dictionary tuning would significantly improve the performance of a system. In the case of the HARPY system developed at Carnegie-Mellon University [Lowerre, 1976], tuning the dictionary was said to have made tremendous improvements in overall performance.

While dictionary tuning can make significant improvements, it would seem that it is better to hold off on major dictionary changes until the development of the APR component has been completed. Once a pronunciation

has been changed, partially alleviating a problem, it is much less likely that acoustic-phonetic knowledge will be incorporated in the APR to account for the phenomenon correctly.

In order to facilitate dictionary tuning, we have developed a program that accumulates all occurrences of a particular orthographic lexical entry and displays them together. In this way, necessary changes to pronunciations become more evident.

4.3 Inconsistencies

There are several errors or inconsistencies in the existing APR program, whose correction would, in some cases, make a significant improvement.

The first is an inconsistency between the allowable pronunciations in the lexicon and the hand labels in the data base used to design and tune the APR algorithms. With respect to the hand labels, a concerted effort was made to insure that they indicated the phonemes which were "really there". On the other hand, the emphasis in designing the dictionary was to insure that the combination of given base forms and the set of phonological rules, would result in all believable pronunciations. When an odd pronunciation was found in the hand labels, we would like to have done one of the following:

a) decide that the pronuncation in the hand labeling was a possible variant of the word and include it in the dictionary.

b) decide that the speaker mispronounced the word, and discard the utterance involved.

c) decide that the hand-labeling was incorrect, and make it conform to an existing pronunciation. This option is dangerous and should only be done as a last resort, because it means that two different phonetic sounds, which were labeled differently because they sounded different and had different acoustic manifestations, are being considered identical. This puts the burden on the APR to identify them as belonging to the same set, although objective evidence indicates otherwise.

A second inconsistency was noticed with regard to the modeling of multidimensional probability densities. In comparing several related probability densities of the same acoustic features, each dimension of the multi-dimensional feature space was normalized by the standard deviation of

that feature within the particular subclass. Though this is basically correct, the normalization should be the same for all members of a class. For example, two of the features used to distinguish [P] from [K] are burst frequency and VOT. (Although the burst for a [P] lies above 10 KHz, the absence of any strong frication leads to a peak appearing in the spectrum between 0-2 KHz. Since VOT depends somewhat on the stress of the following vowel, which is not taken into account, there is a wide range of VOT for both [P] and [K].) The scatter diagram below shows that the cluster for [K] is wholly included within the much larger cluster for [P].

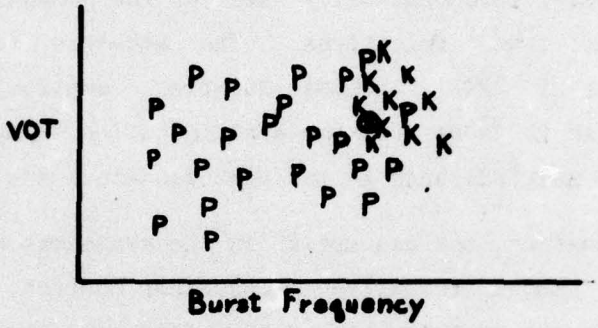


Fig. 14: Burst Frequency vs VOT for [P] and [K].

If the distance from an unknown point (indicated by the circled X) to each of the other known points is measured in terms of the standard deviation of the cluster that includes the known point, then the unknown point shown will seem just as far from the [K] cluster as from the [P] cluster. Many of the [K] samples are, in fact, incorrectly classified as [P]. If the standard deviation of each dimension is taken from the combined set, then this problem is eliminated. Only those few [P] samples that fall within the dense [K] cluster will be misclassified.

When this subtle change was tested using the APEF, correct classification was increased. A third inconsistency lies in the details of the selective modification. While selective modification has improved the discrimination among members of each class of phonemes, there has not been as great an improvement in our overall labeling performance (Section B.3.2). Part of the problem lies in the selective modification of phoneme scores for segments for which those phonemes are only somewhat likely. As described in section B.1.6, for this case, the new score for the phoneme is:

$$\frac{P(\text{ Label-j } | \text{ Phoneme-i })}{P(\text{ Label-j })} * \text{MOD}$$

where MOD is the probability ratio that separates that phoneme from others within its class. For example, if a segment is labeled RETPT ("retroflexed P or T") and the phoneme is [TH] (a fricative), then the equation would be:

$$\frac{P(\text{Label=RETPT}|\text{Phoneme=TH})}{P(\text{Label=RETPT})} * \frac{P(\text{Frication Acoustics}|\text{Phoneme=TH})}{P(\text{Frication Acoustics}|\text{Phoneme=FRIC})}$$

While the label RETPT is occasionally used for the phoneme [TH], it almost never happens for other fricatives. The acoustic measurements that determined the label RETPT include duration, amplitude, and frequency measurements similar to those used for discrimination among the fricatives. Therefore, the two multiplicands in the equation above are not independent.

Even more important, the assumption in the denominator on the right that the phoneme is a fricative is neither correct, nor even likely. Therefore, this segment is being scored as a fricative, even though it was not included in the training data for fricatives. The result has been that more than half of the segments corresponding to the allophone [TG] had [TH] as the top scoring phoneme.

It is felt that a much better formula for this type of modification is:

$$\frac{P(\text{Label=RETPT}|\text{Phoneme=FRIC})}{P(\text{Label=RETPT})} * \frac{P(\text{Frication Acoustics}|\text{Phoneme=TH})}{P(\text{Fric. Ac.}|\text{given FRIC modif. done})}$$

The condition in the denominator on the right indicates that statistics must be gathered for all those cases where it is known that frication modification will be used, data that can be determined from the initial segment label.

Since this would involve a major change, we tried just eliminating the cross-class modification. That is, the score on phoneme [TH] in this case was just

$$\frac{P(\text{Label=RETPT} \mid \text{Phoneme=TH})}{P(\text{Label=RETPT})}$$

This greatly improved the labeling performance measures. The first set of performance statistics below indicates the performance gathered just for the unvoiced plosive allophones.

Percent correct within N choices out of 71.

N	1	2	3	4	5	6	7
	44%	67%	72%	75%	77%	78%	80%

Avg. Log ratio for correct segments 0.58
Avg. Log ratio for incorrect segments 0.83

Ratio of correct sum to incorrect sum 0.54

The peformance data shown below is for the same phoneme/segment correspondences, but without the cross-modification.

Percent correct within N choices out of 71.

N	1	2	3	4	5	6	7
	52%	69%	73%	76%	78%	79%	81%

Avg. Log ratio for correct segments 0.64
Avg. Log ratio for incorrect segments 0.81

Ratio of correct sum to incorrect sum 0.85

The improvement of the ratio measure from 0.54 to 0.85 achieved by eliminating the cross-modification is significant.

4.4 New APR Program

A new version of the APR program has been implemented, incorporating two of the simpler changes suggested above. These are the elimination of the "cross-modification" and the use of LEZ measured at the beginning of fricatives rather than at the "peak" in energy. The overall ratio of the correct sum to the incorrect sum for the new version is 0.92 as compared to 0.78 for the final version of the APR used in HWIM. Though the changes only affect a small percentage of the phonemes, the improvement was substantial.

C. A SPEECH SYNTHESIS-BY-RULE PROGRAM FOR RESPONSE GENERATION AND FOR WORD VERIFICATION

Introduction

The speech synthesis-by-rule program described in this paper is used in two places in the HWIM speech understanding system. First, it is employed in the response generation component to speak the sentence that is composed as a response (Vol. 5, Sec. F). Secondly, and more importantly, it is used in HWIM's Verification component (Sec. D).

In both response generation and word verification applications, the input to the synthesis program consists of a word or sequence of words represented in terms of phonemes, stress markers, morpheme- and word-boundary symbols, and syntactic structure markers. In the response generation application, the output is an acoustic waveform that is played through a digital-to-analog converter, while in the word verification application, a modified version of the synthesis program produces a parametric representation that is to be compared with a parameterization of a portion of the unknown utterance.

This section describes the synthesizer program in its two incarnations and how it interfaces with other components of the system. Techniques for evaluating and optimizing the synthesizer output are also considered. Finally, we present several ideas that were planned, but not implemented in time to be included in the demonstration system.

While previous speech synthesis-by-rule programs for English had as their only objective producing maximally intelligible, natural speech from either a phonemic representation [Liberman et al., 1959; Holmes et al., 1964; Coker, 1967; Mattingly, 1968; Dixon and Maxey, 1968; Klatt, 1972; 1976] or from written text [Coker et al., 1973; Nye et al., 1973; Allen, 1973], word verification requires good spectral matches be produced as a function of time for different talkers. This turns out to be a more stringent requirement, as will be discussed below.

As is now well-known, phonemic information is by itself an insufficient basis for either synthesizing perceptually acceptable

utterances or producing good spectral matches anywhere in a sentence. Acoustic characteristics of words depend also on syntactic and semantic factors. These higher-level constraints influence the durational pattern, fundamental frequency contour, stress pattern, and segmental organization across word boundaries. As will be shown below, these constraints are taken into account in the phonological component of the synthesis-by-rule program.

1. A Model of Sentence Generation

A simplified functional block diagram of sentence production is shown in Figure 1. The figure details our best guess as to what kinds of information should be provided to a synthesis program from a formal linguistic theory. As we will see, not all of this information can be obtained from components of the HWIM system. In the figure, an acoustic output is generated from an abstract sentence representation provided by the semantic, syntactic, and lexical components of the speech understanding system.

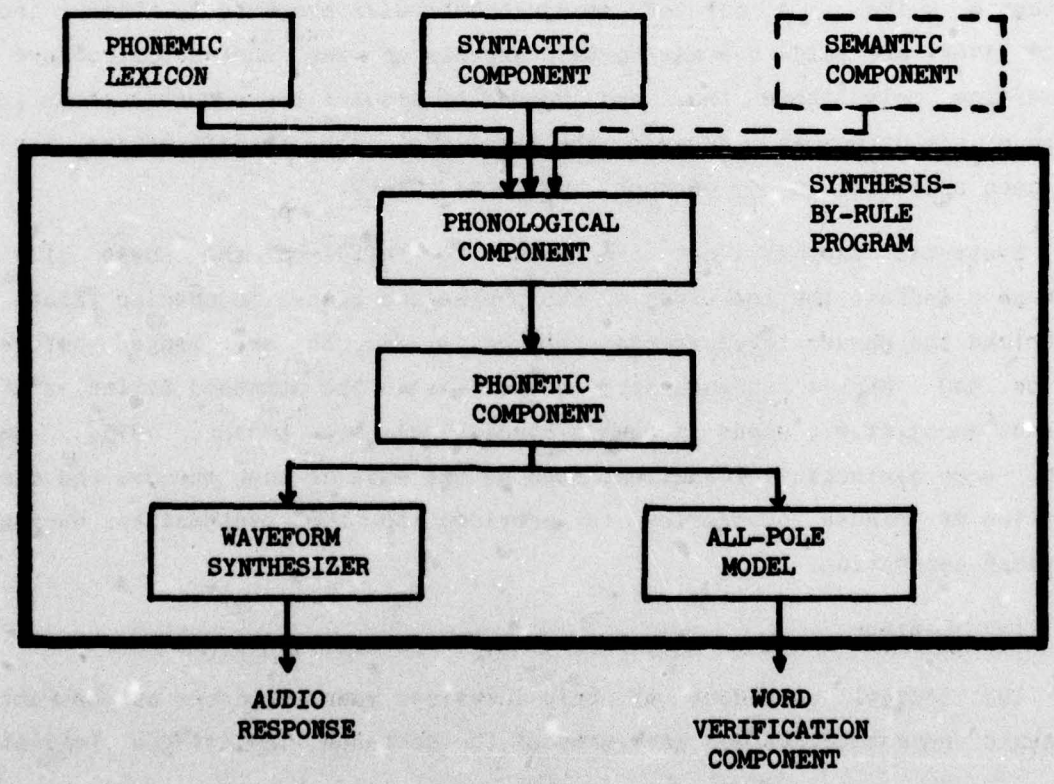


Fig. 1. A functional block diagram of sentence generation.

(a) Semantics

An utterance may contain such aspects of meaning and intention as contrastive stress, emphasis, contradiction, and the speaker's mood. Bolinger [1972] and others suggest that the use of contrastive stress and emphasis is widespread in English, and thus the prosodic shape of sentences cannot be predicted from syntactic considerations alone. For example, the segmental durations and fundamental frequency motions occurring during the production of an adjective-noun sequence change dramatically depending on which word is the information focus [Klatt, 1976b] (N.B. This type of information concerning emphasis and contrastive stress is not provided in the current implementation of HWIM.)

(b) Syntax

The syntactic surface structure of a sentence consists of a complete labeled bracketing of the word string, including a description of the syntactic category of each lexical item and its relation to larger syntactic units. A set of readjustment rules proposed by Chomsky and Halle [1968] are said to erase certain aspects of the surface structure, preserving only those that are needed to predict the acoustic-phonetic shape of the utterance. However, the details of this transformation have not been specified in any current linguistic theory.

Syntactic factors that are known to influence the shape of an utterance include the locations of its phrase and clause boundaries [Klatt, 1976b]and the phrase-level stress pattern. Segments are longer before clause and phrase boundaries, as well as in the stressed syllables of content words, i.e., words in open syntactic classes [Umeda, 1975]. In HWIM, some syntactic information, such as the ends of noun phrases and the location of clause boundaries is provided to the synthesizer during response generation.

(c) Lexicon

The lexical component of this idealized model produces an abstract phonemic representation for each word of the sentence, including a lexical

stress pattern [Chomsky and Halle, 1968]. It is not appropriate for it to produce a detailed phonetic transcription because many phonological rules act both within words and across word boundaries. A more abstract representation, however, permits the model to capture some of the phonological regularities in the lexicon at little extra cost.

In such an abstract representation, the lexical entry for the word "backache" might be given as:

/ # B ′1 AE K * ′2 EY K # /

The slashes indicate an abstract phonemic transcription and will be used only to refer to the lexical forms for words. The "#" signifies a word boundary, and "*" indicates a morpheme boundary. The "′1" identifies a vowel carrying primary lexical stress. The "′2" indicates secondary lexical stress (Chomsky and Halle stress numbers 2 through n), and vowels that are not preceded by a stress mark are unstressed. All of the nonphonemic symbols of this example are needed by the phonological component in order to derive the correct phonetic form for this word. For example, the morpheme boundary is used to prevent a rule from introducing strong aspiration in the release of the medial /K/.

Syllable boundaries are not marked in the abstract lexical representation sent to the synthesis program because it appears that only the presence or absence of a morpheme boundary will influence the allophonic composition of a word. For example, appropriate allophones can be selected for intervocalic consonant clusters without knowledge of syllable boundary locations.

(d) Phonological Component

The semantic information, surface structure and lexical representation serve as input to a phonological rule component, whose output is a less-abstract phonetic and prosodic representation for the utterance to be produced. In this representation, all semantic, syntactic and word boundary information has been removed, yielding a phonetic transcription as well as a complete specification of the prosodic shape for the sentence in

the form of a set of segmental durations, stress levels, and fundamental frequency targets.

Phonetic features are usually considered to be binary, or to have only a few discrete possible values in the output of the phonological rule component. Under this view, only segmental substitutions, deletions, insertions, and feature changes are described by phonological rules.

(e) Phonetic Component

In an ideal model of human sentence generation, rules of the phonetic component transform a string of phonetic segments into sequences of motor commands to the articulators. However, in the phonetic component of the synthesizer, there is no representational level corresponding directly to articulation. Phonetic segment sequences are transformed directly into acoustic variables relating to the shape of the vocal tract and the laryngeal configuration. The acoustic variables (such as formant frequencies, fundamental frequency, and source amplitudes) are used to control a terminal analog speech synthesizer.

The phonetic-to-acoustic transformation is realized by a large set of heuristic rules that have been formulated in a specially-designed programming language [Klatt, Cook and Woods, 1975]. The structure of the phonetic component is such that, for each new phonetic symbol, the next portion of the contour for each variable parameter is determined by either looking up a target value for the parameter in a table or executing a set of rules that compute a target value with reference to manner-of-articulation or place-of-articulation features of the segment.

Target values may then be modified depending on features of the preceding and following phonetic symbols, as well as the stress pattern and durational structure of the utterance. Diphthongs (e.g., the vowel nucleus in the word "bite") involve a sequence of motions between two targets. Complex phonetic segments are also treated in this way. (E.g., plosives like /p/ in "pen" begin with a silent interval during which the mouth is closed, followed by a burst of noise and rapid changes in resonant structure as the stop occlusion is released.)

Transitions between target values are determined by transition time constants that are also computed by rules that take into account features of adjacent phonetic symbols. Most transitions are smooth, moving from one target to another by contours that have a half-cosine shape, but the rules occasionally specify abrupt changes in parameters such as source intensity. If time constants are longer than segmental durations, the smooth trajectories never reach the target. This simulates the behavior of speech articulators that often undershoot an ideal target configuration.

Examples of phonetic feature encoding rules of English include motor reorganization as a function of speaking rate [Gay et al., 1974] and the permitted use of anticipatory nasalization and lip rounding as a function of the phonetic environment. Rules may also have to adjust the relative timing between certain control parameter motions. For example, voicing onset is delayed as much as 80 msec relative to the formant frequency transitions that signal the release of /p,t,k/.

(f) Waveform Synthesizer

An all-digital terminal analog speech synthesizer is employed to compute samples of an acoustic waveform. In HWIM, waveform samples are first saved on disk and then played out through a digital-to-analog converter at 10,000 samples per second, using a 5000 Hz low-pass filter and loudspeaker.

Table 1.

Variable control parameters and synthesis constants that control the terminal analog speech synthesizer shown in Figure 2.

Voicing Source
- Impulse generator
  - F0 fundamental frequency in Hz
- Glottal pulse shaping resonator RG1
  - FG1 = 50 Hz
  - BG1 = 150 Hz
- Glottal pulse shaping resonator RG2
  - FG2 = 0 Hz
  - BG2 cutoff frequency of low-pass filter in Hz*
- Intensity control
  - AV voicing amplitude in dB*

Aspiration and frication sources
- Random number generator, uniform distrib. (-1. to +1.)
- Modulate noise samples by square wave (50% modulation if F0>0)
- Frication shaping resonator
  - FF1 = 0 Hz
  - BF1 = 1200 Hz
- Intensity Controls
  - AH aspiration amplitude in dB
  - AF frication amplitude in dB

Vocal tract transfer function I (glottal sound sources)
- Five cascaded oral formant resonators R1-R5
  - F1 first formant frequency in Hz
  - B1 first formant bandwidth in Hz
  - F2 second formant frequency in Hz
  - B2 = 80 Hz
  - F3 third formant frequency
  - B3 = 110 Hz
  - F4 fourth formant frequency in Hz
  - B4 = 200 Hz
  - F5 fifth formant frequency in Hz
  - B5 = 200 Hz
- Nasal resonator RNP and antiresonator (zero pair) RNZ
  - FNP nasal formant frequency in Hz
  - BNP = 250 Hz
  - FNZ nasal zero frequency in Hz
  - BNZ = 250 Hz

Vocal tract transfer function II (fricatives and bursts)
- Six parallel formant resonators R2'-R6 and bypass path
  - AB attenuation to bypass path amplitude in dB
  - A2 attenuation to second formant amplitude in dB
  - A3 attenuation to third formant amplitude in dB
  - A4 attenuation to fourth formant amplitude in dB
  - A5 attenuation to fifth formant amplitude in dB
  - A6 attenuation to sixth formant amplitude in dB
  - F2' thru F5' have the same values as F2 thru F5
  - B2' thru B5' have the same values as B2 thru B5
  - F6 = 4700 Hz
  - B6 = 250 Hz

Radiation characteristic
- First difference of input time samples

---

*value changed only when glottal impulse issued

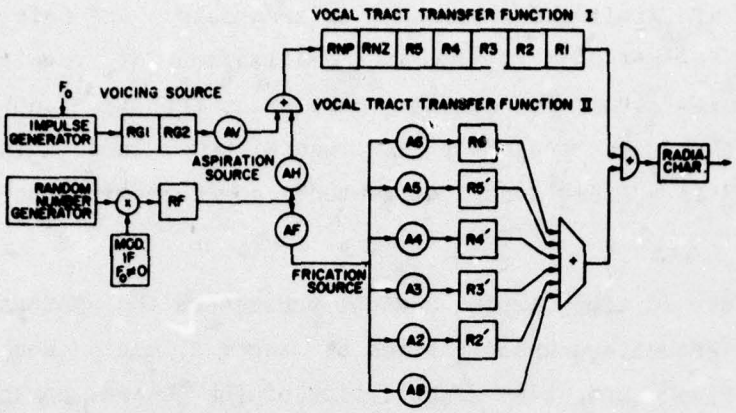


Figure 2. A block diagram of the digital terminal analog speech synthesizer that is used to convert a set of time varying acoustic parameters into an acoustic waveform.

A block diagram of the synthesizer configuration is shown in Figure 2. Variable control parameters and synthesis constants that determine the output characteristics of the synthetic speech are listed in Table 1. Parameters determine (a) the intensity and spectral characteristics of several sources of sound energy (voicing, aspiration and frication), (b) the resonant frequency and bandwidth of a set of resonators that are configured to approximate the resonant properties of the vocal tract and nasal cavities, and (c) the derivation of sound pressure that is radiated from the lips and nostrils.

Each block having a label beginning with the letter "R" is a digital resonator, or second-order difference equation whose coefficients are determined by specification of a resonant frequency and a resonance bandwidth [Klatt, 1972]. Circles having labels beginning with the letter "A" are amplitude controls. The resonators RG1 and RF serve as fixed low-pass filters that give the proper spectral shape to the glottal volume velocity and turbulence noise respectively. RG2 is programmed to act as a variable low-pass filter whose cutoff frequency is high for normally voiced sounds and low for voicebars and voiced fricatives, where the vocal fold vibration pattern is smooth and thus contains little energy in the higher harmonics. The output of the random number generator is modulated by a 50% duty-cycle square wave if the fundamental frequency F0 is non-zero in order to simulate the amplitude fluctuations of turbulence noise due to variations in air flow whenever the vocal folds are vibrating.

RNZ is a digital anti-resonator or zero pair. The pole pair RNP and the zero pair RNZ are used to simulate nasalization of vowels; they are effectively taken out of the cascade vocal tract transfer function if they are given the same frequency and bandwidth values. The radiation characteristic is a first order difference approximation to taking the derivative.

A set of 20 time-varying control parameters and synthesis constants determine the acoustic characteristics of each 5 msec chunk of output waveform. (Five msec time quantization of the control parameter data is fast enough to simulate even the most rapid speech events.)

2. All-Pole Synthesizer

For the verification in the HWIM application, a modified synthesizer computes a simplified all-pole representation of the speech spectrum every 10 msec, using the parametric data produced by the phonetic component of the program (5 formant frequencies, 5 bandwidths, and amplitudes of the voicing source, aspiration source and frication source). These all-pole spectra are then compared with all-pole spectra obtained from the unknown utterance.

For non-nasal sonorants, the spectral representation consists of 3 complex-conjugate pole pairs corresponding to the lowest three formants of the synthesis. This 6-pole representation is compared with formant contours extracted from the unknown speech waveform. Three formant frequencies were chosen as the basis for spectral comparisons for non-nasal sonorants because it is fairly easy to normalize the ranges of the formants for different speakers, and it avoids the difficult problem of prediction the spectral shape above 3 kHz.

For all other speech sounds, a 13-pole spectrum is computed, because formant tracking is either unreliable or meaningless in voiceless portions of the unknown utterance. The 13-pole synthetic spectrum is derived from 5 formant frequency/bandwidth predictions plus three poles that are set so as to model the spectrum of the sound source. Three cases must be considered, (1) voiceless sounds (fricatives, plosive bursts, and

aspiration), (2) voiced nasals, and (3) voiced fricatives. Since the vocal tract transfer function in a fricative or plosive contains zeros as well as poles, the net effect of a transfer function zero is simulated by increasing the bandwidth of those formants that are cancelled by back-cavity zeros (Klatt, 1972).

The Verification component always performs a spectral match on a 6-pole spectrum, so it is necessary to reduce the 13-pole spectrum to a 6-pole model. This is done in the reflection coefficient domain because it is thereby possible to find the locations of 6 poles that give a best match to the 13-pole spectrum.

3. The Phonological Rule System in HWIM

In order to program the rule system that we had envisioned, a special language was needed. There presently exist languages for expressing segmental phonological rules in terms of features [Zue, this volume; Bobrow and Fraser, 1968; Cohen and Mercer, 1974], but these languages are not sufficiently general, as will be shown below. (In addition, since the Zue rules generate forms that reflect idiosyncratic behavior on the part of HWIM's APR component, these forms are not appropriate for the synthesis component.) In our development of a general rule language, we have used ideas of Carlson and Granstrom [1974, 1975] from their design of a general programming language for speech synthesis, and have written a compiler to transform programs written in this language into efficient machine code. (Our compiler produces Fortran code so as to be somewhat more machine independent, but at a cost in run-time efficiency.)

Some of the requirements for a general phonological rule language are suggested in Table 2, which presents an example of the input format to the phonological rule program for the initial portion of the sentence "Buttons are often lost in this machine." (See Table 3 for definitions of symbols.) For example, we need to be able to indicate such syntactic information as the beginning of a content noun (#N in Table 2), the end of a noun phrase (]N) and the beginning of a verbal auxiliary (#VA), as well as such lexical information as the vowel carrying primary lexical stress ('1).

- -

Table 2.

Example of the input to, and the output from the phonological component.

SENTENCE TO BE GENERATED:
"Buttons are often lost in this machine."

INPUT TO THE PHONOLOGICAL COMPONENT:
#N B ´1 AH T AX N Z]N #VA ´1 AA R ...

OUTPUT FROM THE PHONOLOGICAL COMPONENT:

Segmental string	B	AH	GLSTOP	EN	Z	ER	...
Stress	1	1	0	0	0	0	
Duration in msec	80	80	30	120	90	70	
Fund. Freq. in Hz		150	150	120		110	

- -

In the language we have designed, each legitimate input symbol is defined in terms of binary categorization features. Features are employed as a convenient method of referring to sets of input symbols that behave similarly with regard to phonological and phonetic rules. Distinctive features are not used in any other phonetic or articulatory sense in the program.

- -

Table 3.

Notation for defining phonetic segments, stress marks, syntactic symbols, and semantic symbols in terms of features. Comment lines are preceded by "*".

```
:SYMBOLS=(IY,IH,...,´1,´2,´!,...#N,#VA,")N",".",...)

:FEATURES=(SEG,VOWEL,FRONT,...,STRESS,...,WBOUND,...)

*Vowel segment in "beet"
    [IY]=(SEG,VOWEL,FRONT,...)
*Vowel segment in "bit"
    [IH]=(SEG,VOWEL,FRONT,...)
*Primary lexical stress
    [´1]=(STRESS)
*Secondary lexical stress
    [´2]=(STRESS)
*Semantic emphasis
    [´!]=(STRESS)
*Beginning of a content noun
    [#N]=(WBOUND)
*beginning of a verbal auxiliary
    [#VA]=(WBOUND)
*End of a noun phrase
    [")N"]=(PBOUND)
*End of a declarative sentence
```

Examples of several feature definitions are presented in Table 3. Each input symbol, whether segmental, stress, syntactic, or semantic, could be defined uniquely by features, but only those features that are actually used to advantage in particular rules are presently included in the program. The addition of a feature to the program is a simple task, as can be seen by the format shown in Table 3.

The output of the phonological rule program (see Table 2) consists of a sequence of phonetic segments, a stress assignment to each segment, a duration for each segment, and certain phonological aspects of the fundamental frequency contour for vowel nuclei.

(a) Stress

Stress is considered as a segmental attribute at the level of the output of the phonological component. Stressed segments are longer in duration, are produced with a slightly greater subglottal pressure, and are less likely to show reductions in glottal and supraglottal motor commands (as realized here by changes to source characteristics and formant patterns). Stressed vowels of content words (vowels preceded by "1" or "2" in the input text), and prestressed consonant clusters that form a legitimate word initial cluster are called stressed at a segmental level, though the distinction between "1" and "2" stress is eliminated. This distinction is only used in the phonological rule program for computing fundamental frequency contours.

A segmental stress feature is implemented in the output of the phonological component as a separate array, as shown in Table 2. The value for segmental stress is 1 if the segment is stressed and 0 otherwise.

(b) Segmental Duration

The phonological rule program computes a duration for each phonetic segment. Rules modify the basic duration assigned to each segment type depending on the use of emphasis, the locations of phrase and clause boundaries, presence or absence of stress, the locations of word boundaries, and detailed interactions between adjacent segments [Klatt, 1976a].

As duration and fundamental frequency are continuous variables, it does not seem possible to formulate rules concerning them in terms of binary or n-ary features in any natural way. Therefore, it is necessary to augment the phonological rule language in order to be able to manipulate continuous variables.

An example of the notation that has been adopted is illustrated in Table 4. The example contains two parts, a statement of the conditions under which action is to be taken and the durational change to be made. This rule states a simplified version of phrase-final lengthening, which is implemented in this case as non-final shortening: that is, the duration of a segment is to be decreased unless the segment is in the last half-syllable before a phrase boundary. Specifically, segments are shortened if, when the succeeding input symbols are scanned, a syllabic segment is encountered before a phrase boundary symbol [Klatt, 1976a]. If a segment meets this criterion, its duration DURCUR is changed according to the formula shown in Table 4, where DURMIN is the minimum possible duration for the current phonetic segment. Highly incompressible segments such as labial plosives have a larger DURMIN value and are therefore less influenced by this rule than other segment types.

```
["."]=(PBOUND,CBOUND) lm 0
```

Table 4.

Phonological rule with a variable context. The rule will be applied if a segment with the feature +SYL occurs within 8 segments to the right of the current segment and there are no intervening segments with the feature +PBOUND. The rule shortens segments in the initial portion of each phrase.

```
*Shorten segments in a non-phrase-final syllable
    (+SEG)7...(#0 7 (-PBOUND))(+SYL)
        DURCUR = 0.6*(DURCUR - DURMIN) + DURMIN
```

In rules such as that given in Table 4, an indented arithmetic statement or phonological rule will be executed if and only if the input symbol currently being processed passed the phonological conditions appearing above it. Recursion of the indentation convention is permitted up to the length of a single line, thus enabling the programmer to express conditions of some complexity with a minimum of notational effort.

Fortran-like "IF" and "GO TO" statements are accepted as part of the language, and these can be used to jump over portions of the rules if desired. It should be clear that the programmer also needs the power of Fortran "DO" loops and arithmetic instructions to set up the loops that keep track of the identity of the current input symbol and its neighbors. Perhaps these bookkeeping tasks could be done automatically in some future version of the program, but they have not been implemented as yet.

The rule in Table 4 actually tests for 8 possible strings that may satisfy the conditions, i.e., from zero to seven input symbols may exist between the present symbol and a syllabic segment, as long as these symbols do not possess the feature +PBOUND (i.e., plus phrase boundary). The notation is similar to that described in Bobrow and Fraser (1968). Rules that involve variable conditions are useful for the statement of some phonological effects that cannot be expressed in any other way, given our assumption of a single linearized input sequence. However, the computational expense of a variable condition rule is to be avoided wherever possible, for example by ordering a rule after word boundaries have been erased if the rule would otherwise involve an optional word boundary condition.

(c) Fundamental Frequency

The fundamental frequency contour for an utterance is characterized by a sequence of target frequencies that are continuous variables. The phonological component computes fundamental frequency target values for each syllable nucleus, based on information concerning basic intonation contours, special semantic symbols that may be present in the representation of the sentence, and its detailed syntactic bracketing. It appears that unstressed syllables have a single target frequency, but that both an initial and a final target frequency are needed to characterize the fundamental frequency gesture over a stressed syllabic nucleus.

Low-level phonetic effects on fundamental frequency, such as the realization of a glottal stop, or the influence of tongue height, or the voicing feature for consonants, are computed by the phonetic component.

(d) Segmental Insertions, Deletions and Substitutions

In the example shown in Table 3, phonological rules concerned with the segmental structure of a sentence convert the schwa-/N/ of "buttons" to syllabic /EN/, /T/ of "buttons" to a glottal stop, and the /AA R/ of the function word "are" to an unstressed retroflex vowel /ER/. Examples of several phonological rules are given in Table 5 in order to illustrate the format used for describing segmental insertions, deletions and substitutions.

The first rule shown in Table 5 converts a schwa-nasal sequence into syllabic nasal if the preceding segment is a homorganic stop and the segment before that is not an obstruent consonant. The rule is ordered to appear after word boundaries have been deleted from the input string. The rule would apply to words like "button" and "but until" but not to "Boston".

The second rule presented in Table 5 states that the input symbol /T/ is replaced by a glottal stop if the following segment is a syllabic nasal. The rule applies to the word "button" because the rule is preceded by another phonological rule that has replaced the schwa-/N/ with a syllabic /EN/. Phonological rules are ordered in this way in order to block certain types of derivations and so as to be able to state rules as efficiently as possible.

- -

Table 5.

Phonological rule examples.

```
SYLLABIC CONSONANTS:
*In words like "button, metal, cap 'em"
     /(+SONOR)(+ALVELR +PLOSIV)...
          [AX][N]-->[EN]
          [AX][L]-->[EL]
     [AX][M]-->[EM]/(+SONOR)(+LABIAL    +PLOSIVE)...    GLOTTAL   STOP
INSERTION I:
*In words like "button"
     [T]-->[GLSTOP]/(+SONOR -NASAL)...[EN]
*In word sequences like "can't make"
     [T]-->[GLSTOP]/(+NASAL)...(+WBOUND)(+NASAL)
```

Phonological rules that have been developed and implemented in HWIM's synthesis-by-rule program include the duration rules described in Klatt [1976], and a new set of fundamental frequency rules. Segmental phonological rules insert glottal stops, tongue flaps, syllabic consonants, and postvocalic allophones of /r,l/, as well as delete or change the segmental composition of consonant clusters in unstressed syllables, especially in frequently-used function words.

(e) Phonetic Rule Program

Acoustic-phonetic rules that have been developed specify vowel diphthongization (including the schwa-like offglides of lax vowels in the speech of the author), the detailed interactions between /w,y,r,l/ and adjacent vowels, and partial neutralization of short and unstressed vowels. Other rules create inflections in formant trajectories due to changes in formant cavity affiliation (e.g., between /y/ and /a/), insert nasalization in vowels adjacent to nasal consonants, model plosive burst spectra and predict the onset and offset frequencies at plosive release and closure by a system of rules. Algorithms compute voice onset time in consonant clusters [Klatt, 1976], devoice nominally voiced fricatives in some phonetic environments, add an intrusive aspiration segment in some fricative-sonorant clusters, determine the environments in which a voiced plosive is accompanied by a voicebar, implement palatalization of segments adjacent to palatal consonants, and cause the first and second formant trajectories to jump discontinuously at the release of a lateral consonant.

Improvements to the rules are being sought in the following way. Broadband sound spectrograms are made of the synthesis of consonant-vowel nonsense syllables and sentences. Differences observed between the synthesis and the speech of a single talker are then used to evaluate and improve the rule program. Computer-generated spectra are also used when the amplitude resolution of the spectrogram is found to be inadequate. This research strategy shows great promise of converging to a reasonable functional model of the speech generation process.

4. Evaluation and Conclusions

Examples of broadband spectrograms produced by the rule program and by the author are compared in Figure 3. No system tuning was done on this utterance, nor was the string of input symbols modified on the basis of prior comparisons between synthesized and natural speech. One can see that the general spectral match is already fairly good, although many parameters have yet to be optimized, and additional new rules are clearly needed in other cases.

After optimization to match the speech of the author, a second version of the program was created to match the speech of a second speaker (WAW). The necessary adjustment of formant ranges by global scale factors, and the modification of the shape of some frication spectra involved only a few hours of human effort. The improvement in verification performance for this second speaker was dramatic.

Fig. 3. Broadband spectrograms of the sentence fragment "The time of Bill Woods' trip to Boston..." as produced by the author (top) and by the rule program (bottom). Frequency from 0 to 5 kHz is plotted on the vertical axis, time from 0 to 2 seconds is plotted on the horizontal axis, and blackness is monotonically related to the energy in a 300 Hz frequency bandwidth as averaged over about 2 msec. The rule program talks at about a 10 percent slower rate than the author.

5. Future Plans

We had intended to include an automatic procedure for performing speaker normalization of the synthesizer output since it is clear that some sort of normalization is highly desirable. Had the time been available, we would have investigated the following algorithm.

Each talker would be asked to record a single normalization sentence at the beginning of a session:

"She soaps socks."

The normalization sentence is processed in the same way as an unknown utterance. The average spectrum of the /SH/ and the average spectrum of the two /S/'s would be used to adjust the spectral shape of the synthesis for all sibilants, and the formant tracks measured during voiced intervals would be used to determine the range of the first and second formants. These observed ranges would then be used to scale F1 and F2 of the synthesis. (An alternative strategy, suggested by Craig Cook, would be to do something analogous to the automatic template generation performed in the Harpy speech understanding system [Lowerre, 1976].)

D. VERIFICATION

Introduction

Given the results of the Klatt-Stevens spectrogram reading experiment [Klatt and Stevens, 1971], it seems clear that the ability to return to acoustic evidence for verifying word hypotheses is important to correct recognition. Only there can one verify the consistency of acoustic clues with respect to the given word hypothesis. Assuming that phonological and coarticulation processes are described by rules that are generative in nature, we feel that such a verification program must involve an analysis-by-synthesis procedure to overcome inaccuracies present in preliminary phonetic analysis and to take account of the effects of the phonological rules. This is in contrast to an analytic approach to acoustic-phonetic analysis such as we have in our APR program (Vol. 2, Sec. B). The synthesis phase of such a Verification component must be able to transform a broad phonetic transcription into a parametric representation or template suitable for comparison with the acoustic parameterization of an unknown utterance.

With respect to synthesis, we have written a high-level preprocessor to translate a set of phonological and acoustic-phonetic rules into a Fortran compilable synthesis-by-rule program [Klatt, 1976a]. The synthesis program takes into account phonological effects across word boundaries, altering the parameterization according to the context in which the hypothesized word is thought to occur. This allows us to derive in near real-time a parametric representation of any word, given its phonetic transcription.

Speech recognition systems using templates extracted from real speech have achieved impressive results. White [1975] and Itakura [1975] have reported on isolated word recognition applications, and [Bridle, 1974] has described a technique for word spotting. For our particular application, we have chosen to use synthetic templates generated by a synthesis-by-rule program. This choice involved several considerations, which are summarized below.

1) The use of templates extracted from real speech requires storing a parameterization for each entry in the lexicon, which for large vocabularies may require a substantial amount of storage. On the other hand, the generation of synthetic templates requires only the storage of the synthesis program with a relatively small number of parameters.

2) In continuous speech, it is easier to deal with contextual effects of surrounding words by using synthetic templates. This is because we can deal with them at the phonetic level (see Section D.3). These effects are particularly important for short function words.

3) In a multi-speaker environment, a system using real-speech templates will perform best if it is trained on each new speaker for the whole vocabulary. This technique is not practical for systems with very large vocabularies. However, in a synthesis program, it is possible to use speaker dependent parameters (e.g., formant targets), which can be extracted from a relatively small speech sample.

The chief limitation of using synthetic templates is their dependence on the ability of the synthesis program to generate accurate parameterizations. Inadequacies in the program may produce incorrect results in the Verification component as a whole.

In addition to a synthesis-by-rule program, the Verification component includes time normalization and parametric matching programs, which measure the fit of the word template against the unknown utterance. Time normalization is done using a dynamic programming algorithm based on a method first introduced by Itakura [1975]. The algorithm involves a non-linear time warping based on the registration of the error metric, in this case the log ratio of the linear prediction residuals. We have modified Itakura's method to allow limited misalignment in time between the parameterization of the hypothesized word and its hypothesized position in the utterance. In actually computing the distance measure between these two, we sum the magnitude error metric between corresponding segments (frames), the correspondence having already been determined by the time normalization technique.

1. Synthesis-by-Rule

To synthesize the parametric representation of a hypothesized word, we use its phonetic spelling(s) as given by HWIM's phonetic dictionary called VERDICT (see Vol. 3, Sec. 3). This transcription, plus any available contextual information in the form of neighboring phones serves as the input string to the phonological phase of a synthesis-by-rule program.

This phase contains all of the program's phonetic rewrite rules plus a number of acoustic-phonetic rules dealing with phonological effects. After being processed by these rules, a more detailed phonetic transcription enters the phonetic phase of the program where a direct phonetic-to-parametric transformation is performed. The output parametric representation consists of a set of time functions that would ordinarily control a terminal analog waveform generator. The synthetic speech waveform is not needed in this application because matching is done at the spectral level, between spectra extracted from the real speech and synthetic spectra computed from the parameters generated by the synthesis program. Parameters available for use in verification include:

1) three formant frequencies and bandwidths (up to six formants for frication spectra)
2) 13-pole linear prediction spectrum
3) a nasal pole-zero pair
4) amplitudes of voicing, aspiration, and frication sources
5) fundamental frequency.

The parameters currently used in word matching are a subset of those available from the synthesis program. For a more detailed description of the structure of the synthesis-by-rule program, the reader is referred to Section C of this volume.

2. Spectral Matching

Our spectral distance measure is the log ratio of the linear prediction residuals. This metric was derived by Itakura [1975] using maximum likelihood, and alternate expressions for it are given in [Makhoul, 1975]. The metric is defined as:

$$d = \log \frac{\sum_{i=-p}^{p} b_i' R_i}{\sum_{i=-q}^{q} b_i R_i}$$

where {R(i)} are the autocorrelations of the matching signal, {b(i)} and {b'(i)} are the autocorrelations of the predictor coefficients for the matching and the reference signals respectively, and p and q are the orders of the respective predictors. Here, the reference signal represents the

real speech and the matching "signal" is represented by the synthetic spectrum. To compute the measured spectrum from the real speech, we divide the speech signal (sampled at 10 or 20 KHz) into 20 msec windows overlapped every 10 msec. A preemphasized 13 pole LPC analysis (autocorrelation method) over 0-5 KHz is done for each window, and the roots of the polynomial are extracted by a pole solving routine [Makhoul and Wolf, 1972]. Of the complex conjugate pairs, three pairs with frequency less than 3100 Hz and narrow bandwidths are selected by a formant extraction routine (see Sec. A, this volume). This procedure reduces the spectral model to a six-pole model defined from 0-3100 Hz. The actual predictor coefficients are then computed by multiplying the three complex conjugate pairs together and collecting terms assuming a sampling frequency of 6200 Hz. Both this 6-pole spectral model and the full 13-pole model are retained for later use in the matching process. All of this preprocessing is computationally expensive, but it needs to be done only once for the entire utterance and requires no extra computation beyond that already necessary for HWIM's initial acoustic analysis.

The synthesis-by-rule program computes a new set of synthesizer control parameters for every 10 msec of speech. From these parameters, we derive an all-pole model defining the synthetic spectral shape for each frame of the synthesis. At present, three alternative representations are derived, depending on whether the synthesis-by-rule program predicts (1) a non-nasal sonorant, (2) silence, or (3) some other phone type. The reason for this trichotomy is that the information bearing aspects of different phones appear in different parts of the frequency spectrum. For vowels and vowel-like non-nasalized sonorants, we multiply together the pole pairs defined by the first three formant frequencies and bandwidths to produce a 6-pole model for the region 0-3100 Hz. For the silent interval of a voiceless plosive or pause, the synthesis-by-rule program provides a 13-pole silence spectrum. For fricatives and nasals, the spectrum from 0-5 kHz is represented by a 13-pole model computed by the synthesis-by-rule program. The 13-pole models are reduced to equivalent 6-pole models over the same frequency range, by converting to the corresponding set of reflection coefficients, taking the first 6 of them, and computing the

equivalent 6th order linear predictor. The same 13-to-6 pole conversion is applied to the 0-5 kHz model of the real speech also. We do this for reasons of computational efficiency, since the 6-pole model is adequate for representing the spectral shape of non-sonorants.

The reasons for using alternate spectral representations are two-fold. First, the relevant information for non-sonorants lies principally above 3 kHz, while for sonorants, it lies below 3 kHz. Furthermore, for sonorants, the spectral region of 3-5 kHz is difficult to predict accurately, because of interspeaker variability.

During the spectral matching procedure, the type of the synthetic spectral model is checked. If it is a full (0-5 kHz) or reduced (0-3.1 kHz) model, it is matched against the corresponding model taken from the utterance. In the cases where the synthesizer has specified a silence spectrum, the energy of the speech spectral frame is compared against an energy threshold. If it falls below that threshold, the speech frame is defined to be silence also, and a spectral distance of zero is used. If the real speech frame is non-silent, then the spectral match is performed.

Given both the measured spectra of an unknown utterance and the synthesized spectra of a hypothesized word, it remains to define a time registration between them. We use the same minimum prediction residual to do both time normalization and computation of the spectral distance measure.

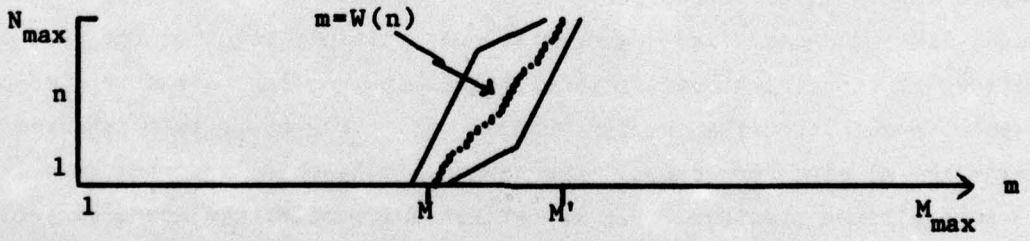


Figure 1

Computing the time normalization means finding the frame-by-frame correspondence between the synthetic and real speech parameterizations. The sequence of these correspondences defines the optimal registration

W(n) as shown in Figure 1. The value of W(n) is the frame number of the real speech which corresponds to frame number n of the synthetic parameterization. This means that for every frame of the synthetic parameterization, there is a frame in the utterance that corresponds to it, but the reverse is not true. There are two sets of constraints that govern computation of the optimal path (Figure 1). These are the boundary conditions and the continuity constraints. In terms of W(n), the boundary conditions are:

$$M-\Delta T_1 \leq W(1) \leq M+\Delta T_1 \qquad \Delta T_1=\Delta T_2=5$$

$$M'-\Delta T_2 \leq W(N) \leq M'+\Delta T_2$$

while the continuity constraints are defined as:

$$\begin{aligned} W(n+1)-W(n) &= \emptyset,1,2 \quad (W(n)\neq W(n-1)) \\ &= 1,2 \quad (W(n)=W(n-1)) \end{aligned}$$

The boundary conditions state that the optimal path must begin and end within specified time intervals around the hypothesized starting and ending points, and the continuity constraints imply that the relative durations of the synthetic and real-speech parameterizations must lie between 1/2 and 2. The open parallelogram shown in Fig. 1 represents the space of all possible time registrations satisfying both sets of constraints.

Itakura [1975] provides a detailed description of the computational procedure for finding the optimal registration. Our procedure is identical to his with one important exception. The endpoints of the registration are made variable to allow for uncertainty in the exact position of a given word. Within limits, the dynamic programming is allowed to find its own optimal alignment. Because of the configuration, all possible time normalizations computed by the dynamic programming are of equal length in terms of the number of frames matched. The program must examine only those paths terminating at the top of the parallelogram to determine which one is optimal (i.e., has the smallest distance).

3. Scoring Philosophy

In order to make the spectral distance scores generated by the Verification component consistent with HWIM'S scoring philosophy, we have gathered performance statistics on a collection of words actually hypothesized by HWIM in the course of its normal operation. Words that were deemed correct as to phonetic spelling and position in the utterance (including embedded words, such as the word "remain" in an utterance containing the word "remaining") were tabulated separately from the list of all words. We then created separate (spectral distance) score distributions for all words and for correct words. The distributions were smoothed, normalized, and entered into the Verifier component as probability density functions. Using these two functions, we can compute log-likelihood scores as the log ratio of the two distributions for a given spectral distance.

This scoring technique produces log-likelihood scores whose dynamic range is considerably less than those word scores returned from the Lexical Retrieval component. (Verification word scores generally fall between +75 and -116, while Lexical Retrieval word scores range from about +350 to below -200.) This difference is due largely to the fact that Lexical Retrieval word scores are based on the sum of individual phoneme scores or likelihoods, while Verification scores are derived from the word taken as a whole. That is, for Lexical Retrieval, correctness is rooted at the phoneme level while for the Verifier, it is rooted at the word level. Since Verifier spectral scores are taken over the whole word, two reasonably long words that differ by only a single phoneme will not have greatly different spectral distance scores nor greatly different log-likelihood scores. In the same case, the Lexical Retrieval component may produce two very different log-likelihood scores if the differing phonemes are ones that the APR almost never confuses.

4. An Example

In HWIM, initial word matching is done at the phonetic level by the Lexical Retrieval component. An example of verifying a word that has been matched by Lexical Retrieval is given below, using the utterance "Give me a list of the remaining trips and their estimated costs" (Figure 2).

```
SIL * R IY M 1 EY N * SIL .
Matched  : Left Boundary Time = 99     Right Boundary Time = 137
Verified : Left Boundary Time = 101    Right Boundary Time = 137
The Verified Score is 724
```

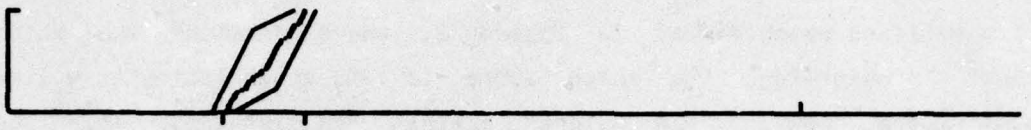


Figure 2

In this example, the phonetic transcription of the word "remain", including its lexical stress, has been sent to the Verification component, along with the frame numbers delimiting the portion of the utterance over which the match is to occur. There may be contextual information in the form of an additional phone at one or both ends of the phonetic input string to be verified. This context is used in computing the synthetic parameterization but does not enter into the matching process.

In the current operation of the Verification component, a phonetic dictionary called VERDICT containing one or more phonetic spellings for each word in the lexicon is used to supply the phonetic input string. Contextual information is obtained by taking the first or last phone of the initial phonetic spelling of the words to be used as context, if any are specified. For the word to be verified, each phonetic spelling in its VERDICT entry is verified in succession with the same context, and the highest scoring match is reported. Here, no context was given, so the synthesis program inserted silence (SIL)* at both ends automatically.

In Figure 2, the duration of the synthetic template is shown as the ordinate while the unknown utterance extends along the abscissa from the origin to the small vertical mark above the line. The two marks below the line locate the hypothesized alignment (frame numbers) as provided by Lexical Retrieval. The parallelogram is open slightly at both ends,

* (SIL) is a dummy symbol implying no coarticulation effects at word boundaries.

allowing the dynamic programming procedure limited freedom in selecting the end points of the optimal registration. In this case, the allowable variations are plus or minus 50 msec.

The computed spectral distance is normalized by duration and subtracted from 1000. An ideal match having zero distance would have 1000 as its verified match score. In Figure 2, where "remain" was matched against "remain"-ing, the match score is 724, which indicates a likely match, where "remain" was matched against "remain"-ing. This score together with the log-likelihood ratio score and the frame numbers delimiting the optimal registration are transmitted back to the Control Component, where they are used to augment the score already computed for the phonetic word match.

5. Results

The spectral score distributions for the Verification component based on approximately 1200 words are shown in Figure 3. These are words that were actually matched by Lexical Retrieval and therefore proposed to Verification during the course of HWIM's normal operation, so they form a biased set acoustically somewhat similar to correct words. They are longer than 100 msec (shorter words are not given to Verification) and were verified mostly without contextual information. (In the final version of HWIM, the synthesis-by-rule program contained a bug that caused it to ignore most contextual information supplied by the Control component.) These were the distributions on which the log-likelihood scores discussed above were based.

Another way of regarding these two distributions is to consider partitioning them at the point where the log likelihood score is zero (where the PDF for correct words equals the PDF for all words). They partition in the ratios shown below:

	LLR≥0	LLR<0
Correct	0.84	0.16
Incorrect	0.34	.66

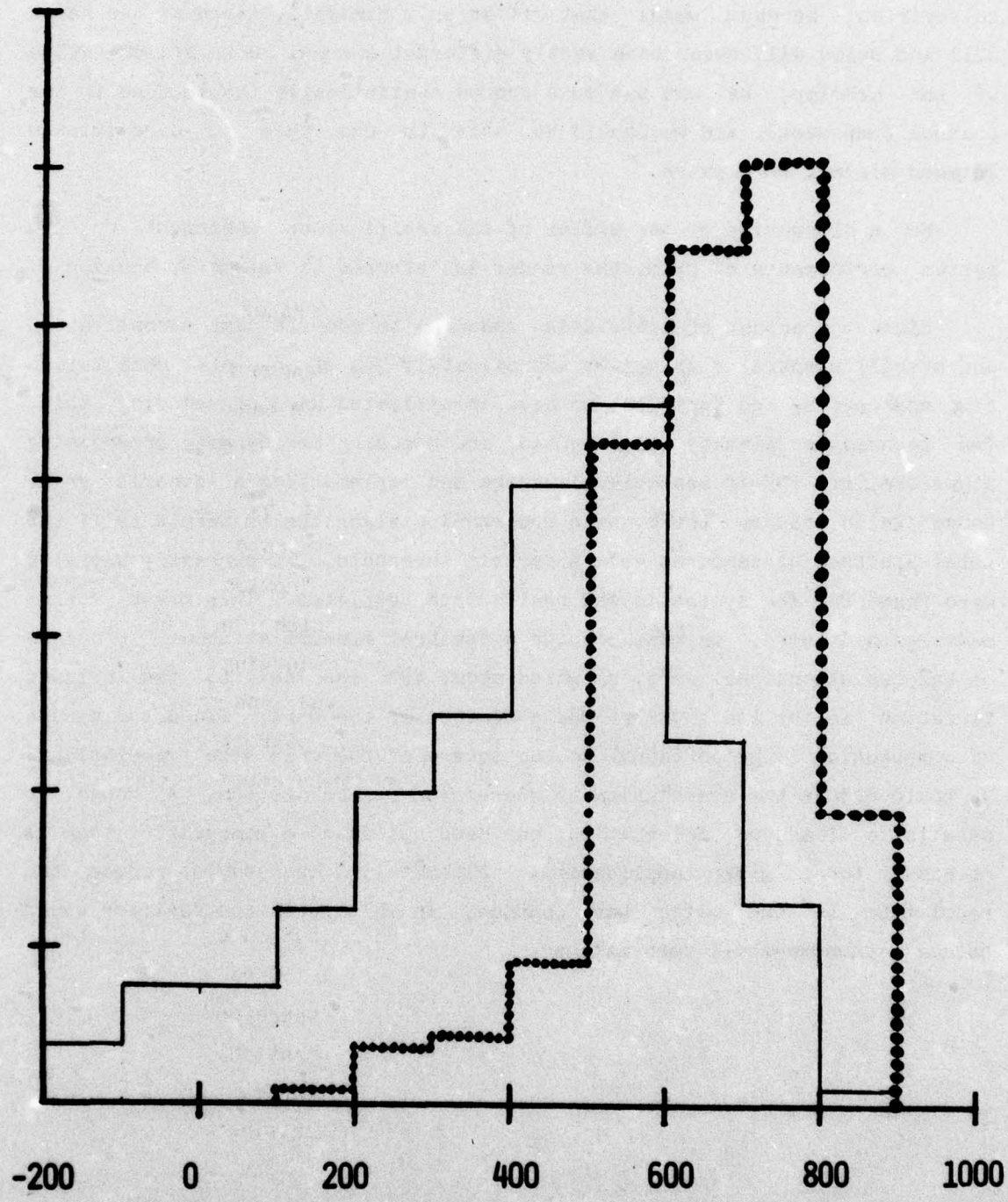


Fig. 3. PDF's of Spectral Scores for Correct and Incorrect Words.

We see that 84% of correct words get a positive score, but so do 34% of incorrect words. We certainly cannot expect ever to achieve a condition of no overlap, because words that differ only minimally (such as the names Bill and Bell) will never have vastly different scores. However, in spite of the overlap, we can use such scores statistically (as is done in the Control component), and we should be able to use them to discriminate between minimal word pairs.

For a discussion of the effect of the Verification component on the system performance of HWIM, the reader is referred to Volume 3, Section F.

Since the amount of computation required to compute time normalization and overall spectral distance is approximately 70% of the total computation time consumed by the Verifier, we have investigated ways of reducing this. Two techniques already implemented are recoding the dynamic programming algorithm into PDP-10 assembler language and implementing a dynamic error bound which causes the dynamic programming algorithm to terminate if the total spectral distance exceeds a certain threshold. We currently use a 10 msec frame for the synthetic and real speech templates. This means for a medium-sized word, we must compute a spectral measure at about 950 points on the two dimensional grid, of which about 400 are due to the allowed variation in the locations of the endpoints of the word. Since the amount of computation is proportional to the square of the grid size (resolution), we could reduce the computation by decreasing the resolution. Although it entails a loss of information, one need not do more computation than is necessary for a given application. Ultimately, one could reduce the resolution to one match per phoneme, in which case the Verifier would become a phoneme-level word matcher.

References

[1] Allen, J. (1973)
"Speech Synthesis from Unrestricted Text," in Speech Synthesis, J.L. Flanagan and L.R. Rabiner, eds.; Dowden, Hutchinson and Ross, Inc., Stroudsburg, Pa.

[2] Bobrow, D.G. and J.B. Fraser (1968)
"A Phonological Rule Tester," CACM 11, 766-772.

[3] Bolinger, D. (1972)
"Accent is Predictable (if you're a Mind-Reader)," Language 48, 633-644.

[4] Bridle, J. and M. Brown (1974)
"An Experimental Automatic Word-Recognition System," Joint Speech Research Unit Report No. 1003, Ruislip, Middlesex, U.K., December.

[5] Carlson, R. and B. Granstrom (1974,1975)
"A Phonetically-Oriented Programming Language for Rule Description of Speech," Proc. Speech Communication Seminar, Stockholm, Aug. 1-3, 1974, Almqvist and Wiksell (in press). See also Speech Transmission Laboratory QPSR 1/1975, 17-26.

[6] Chomsky, N. and M. Halle (1968)
The Sound Pattern of English, Harper and Row, N.Y.

[7] Cohen, P. and R.L. Mercer (1974)
"The Phonological Rule Component of a Speech Recognition System," IEEE Symposium on Speech Recognition, Carnegie-Mellon Univ., April 15-19, 1974, IEEE Catalog No. 74CH0878-9 AE, 177-187.

[8] Coker, C.H., N. Umeda and C.P. Browman (1973)
"Automatic Synthesis from Ordinary English Text," IEEE Trans. on Audio and Electroacoustics AU-21, 293-297.

[9] Coker, C.H. (1967)
"Synthesis by Rule from Articulatory Parameters," Proc. Conference on Speech Communication and Processing, AFCRL and IEEE, Cambridge, Ma., November.

[10] Dixon, R.N. and H.D. Maxey (1968)
"Terminal Analog Synthesis of Continuous Speech Using the Diphone Method of Segment Assembly," IEEE Trans. on Audio and Electroacoustics AU-16, 40-50.

[11] Gay, T., T. Ushijima, H. Hirosi and F.S. Cooper (1974)
"Effect of Speaking Rate on Labial Consonant-Vowel Articulation," J. Phonetics 2, 47-63.

[12] Gillmann, R.A. (1974)
"Automatic Verification of Hypothesized Phonemic Strings in Continuous Speech," System Development Corporation, Report TM-5-315, May.

[13] Gillmann, R.A. (1975)
"A Fast Frequency Domain Pitch Algorithm", J. Acoust. Soc. America 58 Supplement 1, p.S63 (presented at the 90th meeting of the ASA, 5 November 1975, San Francisco).

[14] Holmes, J., I. Mattingly and J. Shearme (1964)
"Speech Synthesis by Rule," Language and Speech 7, 127-143.

[15] Itakura, F. (1975)
"Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans., ASSP, 67-72.

[16] Klatt, D.H. (1972)
"Acoustic Theory of Terminal Analog Speech Synthesis," Proc. 1972 International Conference on Speech Communication and Processing, Boston, Ma., IEEE Cat. No. 72 CHO 567-7 AE, 131-135.

[17] Klatt, D.H., C. Cook, and W. Woods (1975)
"PCOMPILER: A Language for Stating Phonological and Phonetic Rules," BBN Report No. 3080, 18-23, Bolt Beranek and Newman Inc., Cambridge, Ma.

[18] Klatt, D.H. (1975)
"Voice Onset Time, Frication and Aspiration in Word-Initial Consonant Clusters," J. Speech and Hearing Research 18, 686-705.

[19] Klatt, D.H. (1975)
"Structure of a Phonological Rule Component for a Synthesis-by-Rule Program," presented at the 90th Meeting of the Acoustical Society of America, 3-7 November.

[20] Klatt, D.H. (1975)
"Word-Verification in a Speech Understanding System," in Speech Recognition: Invited Papers Presented at the 1974 IEEE Symposium, ed. by D.R. Reddy, Academic Press, 321-341.

[21] Klatt, D.H. (1976a)
"Structure of a Phonological Rule Component for a Synthesis-by-Rule Program," IEEE Trans. ASSP-24, 391-398.

[22] Klatt, D.H. (1976b)
"The Linguistic Uses of Segmental Duration in English: Acoustic and Perceptual Evidence," J. Acoust. Soc. America, 59, 1208-1221.

[23] Klatt, D.H. and K.N. Stevens (1971)
"Strategies for Recognition of Spoken Sentences from Visual Examination of Spectrograms," BBN Report No. 2514, Bolt Beranek and Newman Inc., Cambridge, Ma.

[24] Liberman, A.M., F. Ingeman, L. Lisker, P. Delattre and F. Cooper (1959)
"Minimal Rules for Synthesizing Speech," J. Acoust. Soc. America 31, 1490-1499.

[25] Lowerre, B. (1976)
"The Harpy Speech Understanding System," Ph.D. Thesis, Carnegie-Mellon University.

[26] Makhoul, J.I. (1975)
"Linear Prediction in Automatic Speech Recognition," in Speech Recognition (D.R. Reddy, ed.), New York: Academic Press.

[27] Makhoul, J.I. (1975a)
"Linear Prediction: A Tutorial Review," Proc. IEEE, Vol. 63, No. 4, April.

[28] Makhoul, J.I. (1975b)
"Spectral Linear Prediction: Properties and Applications," IEEE Trans. on ASSP, Vol. ASSP-23, No. 3, June.

[29] Makhoul, J. and J. Wolf (1972)
"Linear Prediction and the Spectral Analysis of Speech," BBN Report No. 2304, Bolt Beranek and Newman Inc., Cambridge, Ma.

[30] Mattingly, I.J. (1968)
"Synthesis-by-Rule of General American English," Supplement to Status Report on Speech Research, Haskins Laboratories, New Haven, Conn.

[31] Nye, P., J. Hankins, T. Rand, I. Mattingly, and F. Cooper (1973)
"A Plan for the Field Evaluation of an Automated Reading System for the Blind," IEEE Trans. on Audio and Electroacoustics, AU-21, 265-268.

[32] Rabiner, L.R., M.R. Sambur, and C.E. Schmidt (1975)
"Applications of Nonlinear Smoothing Algorithm to Speech Processing," Bell Systems Technical Journal.

[33] Schwartz, R.M. (1971)
"Automatic Normalization for Recognition of Vowels of All Speakers," Bachelor's Thesis, MIT, June.

[34] Schwartz, R.M. (1976)
"Acoustic-Phonetic Experiment Facility for the Study of Continuous Speech," Proc. IEEE-ICASSP, April, 1976, pp. 1-4.

[35] Schwartz, R.M. and J. Makhoul (1975)
"Where the Phonemes are: Dealing with Ambiguity in Acoustic-Phonetic Recognition," IEEE Trans. on ASSP, Vol. ASSP-23, No. 1, February.

[36] Tukey, J.W. (1974)
"Nonlinear (Nonsuperposable) Methods for Smoothing Data," 1974 EASCON Record, p. 673.

[37] Umeda, N. (1975)
"Vowel Duration in American English," J. Acoust. Soc. America 56, 434-445.

[38] White, G. (1975)
"Automatic Speech Recognition: Linear Predictive Residual versus Bandpass Filtering," Proc. IEEE International Conference on Cybernetics and Society, September.

[39] Woods, W.A. et al. (1975)
"Speech Understanding Systems, Annual Technical Progress Report," 30 October 1974 to 29 October 1975, BBN Report No. 3188, Bolt Beranek and Newman Inc., Cambridge, Ma.

Appendix 1 - Dictionary Phonemes

The set of phonemes that we use for dictionary spellings is based on the ARPABET, a set of character codes agreed upon by the Data Base Committee at Lincoln Laboratory in March of 1972. The set of ARPABET symbols is shown below.

Phoneme	ARPA	Example	Phoneme	ARPA	Example
i	IY	beat	p	P	pet
I	IH	bit	t	T	ten
e	EY	bait	k	K	kit
ɛ	EH	bet	b	B	bet
æ	AE	bat	d	D	debt
ɑ	AA	Bob	g	G	get
ʌ	AH	but	h	HH	hat
ɔ	AO	bought	f	F	fat
o	OW	boat	θ	TH	thing
U	UH	book	s	S	sat
u	UW	boot	š	SH	shut
ə	AX	about	v	V	vat
ɨ	IX	roses	ð	DH	that
ɝ	ER	bird	z	Z	zoo
aʷ	AW	down	ž	ZH	azure
aʸ	AY	buy	č	CH	church
ɔʸ	OY	boy	j	JH	judge
y	Y	you	ʍ	WH	which
w	W	wit	l̩	EL	battle
r	R	rent	m̩	EM	bottom
l	L	let	n̩	EN	button
m	M	met	ɾ	DX	batter
n	N	net	?	Q	(glottal stop)
ŋ	NX	sing	-		(silence)

To this set, we have added the small number of dictionary phonemes shown below.

ENX - syllabic [ng] as can be in "Washington"
URP - The next six symbols represent unreleased X where X is P,T,K,B,D,G
URT - They can occur when a plosive is followed by another
URK - plosive or a pause.
URB -
URD -
URG -
AXR - Retroflexed Schwa - as can occur in "butter"
UY - Front Rounded vowel (not a diphthong)
EA - "raised" AE, as in "candy"
UAX - Unvoiced schwa as can be in "want to go"
TX - Flapped nasal as can be found in "winner"
UIX - Unvoiced IX as in "multiply"
ST - Context Dependent Allophone - (+strid)(T)(vowel)
TV - Context Dependent Allophone - (-strid)(T)(vowel)
TG - Context Dependent Allophone - (T)(WRLY)
TS - Context Dependent Allophone - (T)(+strid)
LIH - Context Dependent Allophone - IH preceded or followed by L
RIH - Context Dependent Allophone - IH preceded by R
INX - Context Dependent Allophone - IX followed by NX
YN - Context Dependent Allophone - N preceded by [IY,EY,ER,AXR,R]
YM - Context Dependent Allophone - M preceded by [IY,EY,ER,AXR,R]
KA - Context Dependent Allophone - K followed by glides or back vowels
TCH - The frication part of the affricate CH

Appendix 2 - List of APR Labels

The table below gives the complete list of labels which the APR can assign to a segment. Though many correspond to our dictionary phoneme symbols [see Appendix 1], it should be remembered that they are not from the same set of symbols and serve only to indicate a vector of our long-term confusion matrix which is subsequently modified. For those labels that are not like one of our dictionary phonemes, an indication is given as to which phoneme it is intended for.

VOWEL -	vowel in which the formants had some irregularity or abrupt change
SCHWA -	Schwa in which " " "
GLIDE -	intervocalic WRLY but not labeled as such (so might be DH,HH,DX)
IVSON -	intervocalic sonorant (HH,WRLY,DH)
NASAL	
FLAP -	Flapped T or N
IVOBS -	intervocalic obstruent (V,DH,HH,D)
NONNAS	
PLOS -	plosive - usually with a FRIC near it so can't tell if voiced
URPLOS -	unreleased plosive (based on duration)
UVPLOS	
VPLOS -	Voiced plosive - only when no preceding FRIC or following OBS
RETPLS -	retroflexed unvoiced plosive (based on long weak aspiration and possible [R] following
OBS -	non-strident obstruent- above -5 dB
PAUSE -	pause between words
EY	
AY	
AW	
OW	
OY	
IY	
IH	
EH	
AE	
AH	
ER	
AA	
AO	
EL -	syllabic L
UH	
UW	
AX	
AXR	
IX	
Y	
R	
L	
W	
WL -	W or L
T	
TK -	T or K
P	
PK -	P or K
K	
PT -	P or T
KG -	K or G
G	
PB -	P or B
B	

BV -	B or V
V	
F	
JH -	as in Judge
CH -	as in Church
JHT -	JH or T
JHTK -	JH or T or K
JHZH -	JH reconstructed from a ZH (not or ZH)
CHJHK -	CH or JH or K inserted by rule at the beginning
HHQ -	HH or Q inserted at beginning
RETK -	retroflexed K
URKG -	unreleased K or G
BDVDH -	Non-Velar voiced Plosive (e.g. B or D or V or DH)
S	
Z	
SZ -	S or Z
SHZH -	SH or ZH
SH	
ZH	
DH	
FTHDH -	F or TH or DH (or HH)
FTH -	F or TH (or HH)
VDH -	V or DH
NVOBS -	Non-velar (non-strident) obstruent (e.g. V,DH,B)
NVPLOS -	non-velar plosive - usually voiced
MN -	M or N
NX	
M	
N	
RETPT -	retroflexed P or T
NVURP -	non-velar unreleased plosive
BADFRM	
SYLNAS -	syllabic nasal - based on no matching vowel and low F1
BADVWL -	Vowel whose formants and duration didn't match any vowel (usually a segmentation error)

Appendix 3 - APR Rules

This appendix contains a step by step description of the Acoustic-Phonetic Recognition (APR) program as well as a brief description of the rules used in each step. Each rule is implemented as a BCPL program that examines the segment lattice and the acoustic parameters and makes changes in the segmentation or partial labeling. The acoustic parameter names are defined in Section B.

Procedure

1. First, the APR program reads in several general data files:

   a) a dictionary phoneme/segment label confusion matrix. (e.g., <FRONTEND>MATRIX.922)

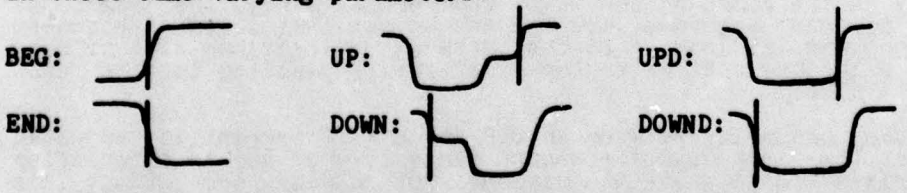
   b) a file with information about vowels. For each vowel, glide, or diphthong, there are three target formant frequencies (normalized by a default fundamental frequency of 231 Hz) and three sets of minimum and maximum duration pairs. These latter values indicate the limits on the duration of the vowel in the first syllable, last syllable, or any other syllable of an utterance. (<FRONTEND>FORMANT-TARGETS)

   c) files containing the probability density distributions used in the selective modification phase of the program (Section C). The files currently used are:

```
<FRONTEND>
BRES.NX/YNM/NM-MIND;1
.MORN-W/Y;1
.NVMORN;1
.JH/UVP;1
.UVP-W/O-STR;4
.UVP-PREC-AG/EACH/OTHER;3
.FRONT-6-F1/F21/NDUR/MF2/F1D/F2R;4
.FRIC-CM75/ROD/DUR/LEZ0;1
```

2. Then the program reads in all the necessary acoustic parameters for the given utterance.

3. A general dip detector is run on each of the three smoothed wide band energy parameters (LEZ, MEPZ, HEPZ), marking six different kinds of events in these time varying parameters:



The dip detector uses a different set of control values for each of the energy parameters. These specify thresholds, such as the minimum depth of a dip.

4. The program finds the last two consecutive 10 msec frames in an utterance with (ROP > -5.0) & (ZC < 25). This point is considered the last possible point for the end of a sonorant sequence.

5. The results of the dip detector are merged to form a preliminary segment lattice (MERGE).

a) Regions between LEZ-DOWND and LEZ-UPD pairs (i.e. dips found in the parameter LEZ) are called OBS (for obstruent). Other regions are labeled VOWEL (for vowel and sonorant regions).

b) Dips in MEPZ that are within VOWEL region with at least two frames of the VOWEL region on either side are called IVSON (for intervocalic sonorant). If the left or right boundary of the dip corresponds to an LEZ-DOWN or LEZ-UP respectively, the dip is called IVOBS (for intervocalic obstruent - e.g., a weak voiced obstruent, such as [V,DH,DX,HH]). If the MEPZ dip is preceded by an MEPZ-DOWN also within the VOWEL region, the dip is called IVOBS. If instead, the dip is followed by an MEPZ-UP within the VOWEL, it is called NONNAS (for nonnasal intervocalic sonorant). Short (3 frames or less) IVSON segments are relabeled as FLAP (for Flapped Dental).

c) Within a VOWEL region, an MEPZ-DOWN defines the beginning of a SON region (non-vowel sonorant). These regions end either at the next MEPZ boundary or at the next LEZ-DOWND.

d) Portions of the VOWEL regions which haven't been segmented off and relabeled are still named VOWEL. If the resulting segment is less than or equal to 4 frames (40 msec), then the label is changed to SCHWA.

e) OBS regions are split up into FRIC (fricative) and OBS regions. Any interval during the original OBS region that is within a HEPZ-dip (DOWND-UPD pair) remains labeled as OBS. The remaining sections are segmented and labeled as FRIC. To avoid timing effects between LEZ boundaries and HEPZ boundaries, a two-frame leeway is allowed on either end of the original OBS region defined by LEZ.

The resulting preliminary segment lattice has no branching, with regions segmented according to broad acoustic similarity.

6. Four rules are applied to complete the preliminary segment lattice.

a) If the lattice ends with a VOWEL or SCHWA segment less than 6 frames in length, this segment is deleted. (Rule: vatend, for vowel at end)

b) The times of the start of the first sonorant sequence, the start of the last sonorant sequence, and the end of the last sonorant sequence are determined by looking at the lattice. These times will be used later to determine whether to look for effects peculiar to the ends of an utterance.

c) Any boundary occurring between an OBS and a FRIC segment is adjusted to be at the first negative second derivative of ROP to occur after ROP has risen 7.0 dB above the minimum ROP during the OBS. This formula finds the location of a plosive burst if there is one. (Rule: AdjustPlosFricBdry)

d) If an MEPZ dip is completely within a remaining VOWEL segment, it is labeled SCHWA. If the dip is three frames or shorter, then it is labeled as FLAP. The remaining segments are labeled either VOWEL or SCHWA depending on their duration. (Rule: hfeloop)

7. Six rules are then applied to take care of sentence end effects.

a) If the last original LEZ sonorant sequence ends with a VOWEL and a SON, such that the VOWEL is greater than 40% of the duration of the two, the two segments are bridged with a single VOWEL segment. (Rule: bridgefinalSON)

b) If the last sonorant sequence ends with two SON segments, they are merged. (Rule: sonson)

c) If the utterance appears to end with a sonorant sequence, the minimum drop time is checked for a drop of at least 21.0 dB in ROP to a level below -4.0 dB. If this drop time is short enough (less than 8 frames), an URPLOS (unreleased plosive) is added at the end. If the drop time is greater than 5 frames, the URPLOS segment is made optional by bridging it with a PAUSE segment. (Rule: urplosatend)

d) If the utterance starts with a FRIC-OBS sequence, such that the FRIC is less than four frames long, then the sequence is deleted under the assumption that the short FRIC segment must have been due to a non-speech lip smack before the utterance. This is because no English word begins with a short fricative (or any phoneme that might look like one, such as an unvoiced plosive) followed by an obstruent. (Rule: lipsmack)

e) If the sentence appears to start with a VOWEL or a short (less than 7 frames) FRIC followed by a VOWEL, then MEPZ is checked to see whether it first comes up out of the noise more than two frames before the VOWEL. The beginning of MEPZ is defined as the first frame for which the average of three consecutive frames (starting with the frame in question) is 10.0 dB more than the minimum MEPZ in the noise at the beginning of the utterance. If there is no decrease in HEPZ between the beginning of MEPZ and the VOWEL, then this period is segmented and labeled as HHQ, since it usually corresponds to a sentence initial [HH] or glottal stop. If this region is less than four frames long, it is made optional by bridging it with PAUSE. (Rule: hhq)

f) If the utterance appears to start with a short (less than 7 frames) FRIC followed by a VOWEL, an UVPLOS (unvoiced plosive) is proposed, with the burst starting at the beginning of the FRIC segment. During most of the APR program, unvoiced plosives are represented as two distinct segments: a segment corresponding to the silence labeled SI, followed by a segment corresponding to the burst and aspiration labeled as UVPLOS or with some more specific unvoiced plosive label (e.g., P, T, K). These pairs of segments are merged into single unvoiced plosive segments near the end of the program. (Rule: proposeuvplos)

8. Rules are applied to split or merge FRIC and OBS segments and possibly relabel them.

a) For all OBS-FRIC-VOWELorSCHWA sequences, ROP is checked for a dip within the FRIC region. This is defined as ROP rising above -5.0 dB, then falling below it, then rising above again, all within the FRIC. This often indicates two adjacent unvoiced plosives which were both released. If this is found, the FRIC is split optionally into three segments: FRIC-OBS-FRIC. (Rule: MissedBurst)

b) For all IVOBS-FRIC sequences, the two segments are replaced (non-optionally) with a single FRIC. (Rule: ivobsfric)

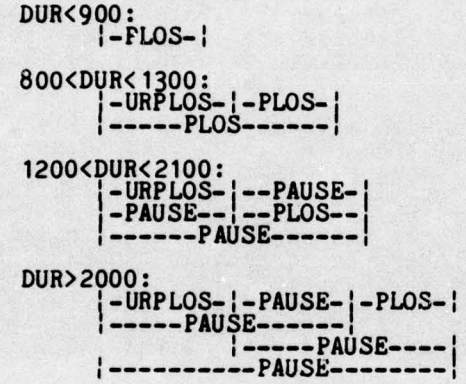
c) If there is a FRIC-OBS-VOWELorSCHWAorSON sequence, such that the OBS is less than 4 frames long, the FRIC-OBS pair is bridged with an optional FRIC. This is to take care of the energy dip which often occurs between frication and a sonorant as in the word SMALL. (Rule: bridgefric)

d) If a FRIC segment is preceded by something other than OBS or PAUSE but has a minimum ROP less than -5.0 dB in the beginning, it is split non-optionally into a PLOS-FRIC sequence. If the FRIC was at the end of the utterance, and the maximum ROP during the FRIC was less than -5.0 dB, it is deleted. (Rule: addobstruent)

e) The "missedburst" rule (see 8a) is again applied.

f) The average values of ROP for the left and right halves of a FRIC segment are compared. If the lower half is not next to an OBS or PAUSE segment or the end of the utterance and the change is greater than 6.0 dB, the FRIC is split into appropriate OBS and FRIC segments. If the change is greater than 10.0 dB, the split is made non-optional. (Rule: splitfric)

g) An OBS segment with minimum ROP less than -5.0 dB is renamed PLOS. Then, depending on its duration, it is split into URPLOS, PLOS, and PAUSE segments according to the criteria below where DUR is the duration in msec. (Rule: UrplosPause)



h) A PLOS-PLOS sequence is changed to an URPLOS-PLOS sequence. This rule is normally unnecessary, unless the two PLOS segments were created by two different rules. (Rule: plos-plos)

i) If an OBS segment with VOWELorSCHWA on both sides is longer than seven frames, it is split optionally into WKFRIC-OBS. This will occur for sequences like HAVE BEEN where the [V] has very low energy due to the following plosive and is noticed as a greatly lengthened silent region. (Rule: splitobs)

j) Each (OBSorPLOS)-FRIC pair is checked for a following VOWEL or SCHWA or the end of the utterance. Any of these would cause the program to expect a long aspiration for an unvoiced plosive. If the FRIC is short and weak enough, the sequence is assumed to be an unvoiced plosive. Consequently, the PLOS or OBS segment is bridged with a SI segment, and the FRIC segment, with an UVPLOS segment. If the FRIC portion is longer than 5 frames, a RETPLS label (retroflexed plosive) is used instead of UVPLOS. If the first segment was labeled PLOS, and the FRIC is very short or very weak, the change is made non-optional. (Rule: uvplos)

k) If a FRIC that is not after the last VOWEL in a sentence is longer than 13 frames, then it is broken into two FRIC segments. If it is longer than 15 frames, this split is made non-optional by deleting the original FRIC segment. (Rule: gemfric)

l) The AdjustPlosFricBdry rule (see 6c) is re-applied to seek bursts within PLOS segments. If one is found more than two frames before the end of the PLOS segment, then the PLOS is bridged with an optional SI-UVPLOS sequence. If none is found, and there is no FRIC segment before or after the PLOS segment (which would cause the VOT

of an unvoiced plosive to be greatly reduced), then the PLOS is renamed VPLOS (voiced plosive). (Rule: vplos)

m) UVPLOS segments are relabeled as either P, T, K, using burst frequency (CM75 measured at the burst), VOT, burst energy (ROP), minimum ROP during the silence, aspiration frequency (CM75 measured in the middle of the aspiration), and F0. Several combination labels such as PK, TK, JHT are used when the decision is not clear. This label replaces the UVPLOS label, leaving the SI segment unchanged. (Rule: LabelUVPLOS)

n) A segment labeled FRIC or WKFRIC is relabeled as one of many more specific fricative labels depending on the frication frequency (CM75) and the total energy (ROP) averaged over the center half of the FRIC segments, along with average F0, duration, and whether there are VOWEL's adjacent to the segment. (Rule: LabelFRIC)

o) If a segment labeled ZH or SHZH is not preceded by a vowel, then it might be the second half of the affricate [JH], since [ZH] must be preceded by a vowel in English. The preceding segment and the SH or SHZH are therefore bridged with a segment labeled as JHZH (indicating a JH caused by noticing a ZH). If the original fricative label was just ZH, it is deleted. If this means there is no right-going path for the preceding segment(s), they are also deleted. If the label was SHZH, it is changed to SH. (Rule: makeJH)

9. The next set of rules is applied to detect and label nasals and intervocalic glides.

a) Any IVSON segment which is longer than 9 frames is optionally split into a SON-IVOBS sequence. (Rule: splitIVSON)

b) This rule looks at all IVSON, GLIDE, FLAP, NONNAS, and SON segments. The first three types are always intervocalic, NONNAS may be intervocalic, and SON is required to be postvocalic. First, the point of maximum decrease of F1 is found in the vicinity of the left boundary. The minimum value of F1 during the nonvowel sonorant is also found. Nasals are marked by a sudden drop in F1 (usually more than 70 Hz in one frame) and a very low F1 throughout most of the nasal (usually less than 300 Hz). Analogous checks are made on the right boundary if the segment is followed by a vowel. If thresholds on the minimum F1 and the maximum change in F1 are met, then the boundaries are moved to the points of maximum change in F1, and the label is changed to NASAL. If the segment was originally SON and was not changed, then it is deleted, and the preceding VOWEL expanded. If the segment is intervocalic and was not changed to a NASAL, then F2 and F3 are searched for valleys, which would correspond to intervocalic [W, L, or R]. If a valley is found in F2, then the segment may be relabeled as W or L. If a valley is found in F3, then it may be relabeled as R. If the label is changed, then the boundaries are shifted to the point where the formant with the valley has risen one third of the way from the bottom of the valley. (Rule: ivson)

c) SON segments which are followed by VOWELorSCHWA segments are examined in a similar manner to determine whether they are nasals. If they are not renamed NASAL, they are deleted. (Rule: prevson)

d) Nasals which were not found as dips in MEPZ or HEPZ in the first phase of the segmentation are detected by this rule. First, if the maximum F1 during a VOWEL region is less than 336 Hz, then it is bridged with an optional SYLNAS segment (for syllabic nasal). If the maximum F1 is less than 280 Hz, this bridge is non-optional. Otherwise, the VOWEL segment is searched for regions where F1 drops below 336 Hz, and sharp jumps in F1 occur as described in rule "ivson" above. These regions are optionally segmented off as

NASAL's, with the remaining regions being labeled as VOWEL or SCHWA depending on duration. If more than one such region is found within one VOWEL region, then all permutations of optional paths will be included. (Rule: findnasal)

e) If the SCHWA in a NASAL-SCHWA sequence is shorter than 3 frames, then the two segments are optionally bridged by a NASAL segment. If the SCHWA is only one frame long, then the change is non-optional. This is necessary, because sometimes the first formant in a nasal will seem to rise preceding an obstruent, causing the program to find a SCHWA between the nasal and the obstruent. (Rule: bridgeSCHWA)

f) Each intervocalic NASAL is examined for a dip in ROP towards the end of the segment. If found, the NASAL is optionally split into NASAL-VPLOS (e.g., as in "AMBER", "UNDER"). If it is followed by a SCHWA segment and is longer than seven frames, it is optionally split into a NASAL-NASAL sequence. This is done because single nasals followed by reduced vowels are usually quite short, and sometimes even flapped if the nasal is an [N]. (Rule: splitnasal)

g) For each segment whose label corresponds to several phonemes, some of which could have the feature +VELAR, and which is preceded by VOWEL or SCHWA segments, the minimum difference between F3 and F2 in each of the 3 preceding frames is measured. (e.g., the label PLOS is intended for all the voiced and unvoiced plosives, and the phonemes [K] and [G] have the feature +VELAR.) If any one of these differences is less than 500 Hz, then the label on the segment is changed to the corresponding +VELAR label (e.g., NASAL => NX, PLOS => KG). If the minimum F3-F2 difference is greater than 500 Hz, the segment is relabeled to indicate a nonvelar consonant (e.g., NASAL => MN, PLOS => NVPLOS, URPLOS => NVURP). If the normalized minimum F3 in the three frames preceding the boundary was below 2600 Hz (2100 Hz for a male with an average F0 of 120 Hz) indicating retroflexion, and if the segment was originally NASAL, then it is changed to MN, since [NX] may not be preceded by any retroflexed phoneme ([R,ER,AXR]) in English. (Rule: velar)

h) For all MN segments, the one-frame derivatives of F2 and F3 are measured at a point two frames before the left boundary of the MN. (The derivative of any parameter at a particular frame is taken to be the difference between the value of that parameter one frame before, minus the value of that parameter at the frame in question.) F2 and F3 tend to rise for dental consonants, and fall toward labial consonants. Therefore, if the sum of these two derivatives is less than -100 Hz, indicating falling formants, the MN is changed to M. If the sum is greater than -50 Hz, the label is changed to N. (Rule: MorN)

i) When a word whose first phoneme is a stressed vowel follows a silent period (as at the beginning of a sentence), there is often a glottal onset before the vowel. This is often recognized as an HHQ segment. In order to allow the word matcher to skip over the HHQ segment, a PAUSE segment is inserted at the beginning of the lattice if it starts with an HHQ. In addition, the program often does not detect sentence initial [DH]. Therefore, if the lattice begins with a VOWEL, SCHWA, or SYLNAS segment, a PAUSE segment will also be inserted, in order that the word matcher can apply the word boundary rule which allows the initial [DH] to be missed in the context of a silence. (Rule: addpause)

j) When the data base comprised only 100 utterances, there were nine 3-1 segmentation errors, i.e. regions where there were three segments in the lattice for only one phoneme in the real speech. Five of these errors were due to a short dip in energy during the middle of a long vowel. This dip had been labeled as FLAP and not relabeled as a NASAL or intervocalic glide. The exact duration of the dip, and the depth of the dip did not seem to correspond to whether it was phonemic or not. Therefore, all VOWEL-FLAP-VOWEL sequences are optionally bridged with a long VOWEL segment. Since the probability of this rule being needed is very low, a later rule will put a low path probability on the bridging segment. (Rule: bridgeFLAP)

10. The following rules are applied to segment and label vowel sequences. At this point, many of the procedures are less like rules and more like long subroutines, in that rather than looking through the segment lattice for an appropriate context, they are called to process a particular region. Unlike rules, the subroutines can call each other recursively.

a) The formant tracker often confuses the formants during sentence initial [W] or [L] due to F1 and F2 being very close together. This is detectable, however, since if this mistake occurs, F2 (and F3) appear to fall several hundred Hz in a single frame, which is impossible for real formants. Therefore, if F1, F2, or F3 drop by more 200, 500, or 300 Hz respectively then the formants are smoothed. (Rule: Wglitch)

b) The function "DoVowels" (see below) is called in order to further segment and label all VOWEL and SCHWA segments. (Rule: vowel)

c) Three possible segmentations are considered for each VOWEL region:

i) The region can be treated as a sequence of one or more vowels.

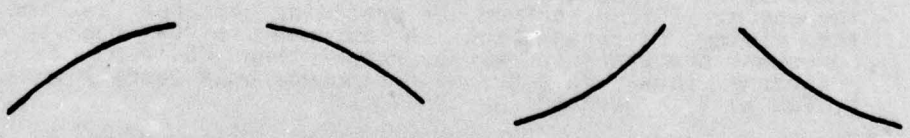
ii) There could be a prevocalic glide, followed by a sequence of one or more vowels.

iii) The region could be a sequence of one or more vowels ending with a postvocalic glide.

The remainder of the subroutines listed in this part are called to perform the particular tasks involved in segmenting and labeling a proposed vowel or sonorant region. (Subroutine: DoVowels)

d) The subroutines "CurveType", "GetTargets" and "VowelSequence" (see below) are called sequentially, and formant glitches found by "CurveType" are handled specially. (Subroutine: EvVowelSeq)

e) Each formant track within a sonorant region is modeled as one of four different curve types:



or as an arbitrary sequence of rising, falling and level sections. Unreasonable formant glitches are also detected. (Subroutine: CurveType)

f) Formant target regions are defined where vowel formant measurements are most likely to be reliable. This is done using the above formant tracks. (Subroutine: GetTargets)

g) Each vowel target region in a vowel sequence is labeled by calling "MakeVowelSeg". If the vowel region is less than 10 frames long, then there can be only one vowel in the sequence, and formant values at the center of the vowel are used. Several other possible segmentations are also considered:

i) If the region was split up, and the first vowel was labeled as a vowel which can't be followed by another vowel in English (e.g., IH,EH,AE,AH,AA etc), then an attempt is made to relabel the whole sequence as a single vowel.

ii) If two adjacent vowels in the sequence receive the same label, an attempt is made to merge them.

iii) IY-laxvowel or EY-laxvowel sequences such as IY-AX in "Give me a list" are detected separately, since there is no clearly defined target region for the lax vowel in this context.

iv) F2 and F3 are examined for possible intervocalic [W,L,R] segments that weren't found by the dip detector and "ivson". If they are found, the two remaining regions are processed recursively by calling "DoVowels". (Subroutine: VowelSequence)

h) Given a set of vowel targets and boundaries, plus a specification of which vowels or glides are to be tried (if not all) the vowels are scored and rank-ordered by the subroutine "VOWELS" (see below). Then the subroutine "CheckOut" (see below) is called for each vowel in the list to see if everything else about the vowel is consistent. If no vowels are found, then the segment is either labeled SYLNAS (if F1 is low) or BADVWL. (Subroutine: MakeVowelSeg)

i) The vowels are scored by first normalizing the formants by pitch [Schwartz, 1971], and then comparing the normalized data with the model formants for each vowel, diphthong, and glide. Seven distance measures (F1, F2, F3, F2/F1, F3/F1, F3-F2, F2-F1) are used to score the choices, which are sorted in order of increasing distance. (Subroutine: VOWELS)

j) The duration of each sonorant is examined in context. For several glides and vowels, checks are made on formant transitions, minima and maxima, etc., for possible inconsistencies with that label. For instance, for AY, F1 must fall and F2 must rise sufficiently. (Subroutine: CheckOut)

k) Each sonorant sequence is checked for the existence of a prevocalic [R] by looking for the rising F3 transition. If the previous label was RETPLS, then the thresholds are more lax, since part of the [R] transition is often unvoiced. "PreVocGlide" (see below) is called if a prevocalic [R] looks likely. [AXR-AX] sequences are also looked for specifically, since they are not usually realized as two distinct target regions. (Subroutine: PreVocR)

l) Prevocalic [L] and [W] are detected by a rapidly rising second formant, in a manner similar to that used in rule "PreVocR". If the energy (ROP) rise from the preceding consonant is too sharp, the rising F2 transition is assumed to be due to a labial consonant preceding the vowel, rather than [L] or [W], so the preceding label is narrowed to include only labials (e.g., MN => M, VDH => V). (Subroutine: PreVocLW)

m) Prevocalic [Y] is detected as a rapidly falling F3, and narrowing F2-F3 distance. (Subroutine: PreVocY)

n) If the preliminary tests for a prevocalic glide are successful, then the three formants for the proposed glide are checked for consistency by calling "MakeVowelSeg". An optional glide segment is then created and "DoVowels" is called to process the remainder of the sequence. (Subroutine: PreVocGlide)

o) Postvocalic [R] and [L] are detected and processed in a manner analogous to that for the prevocalic glides. (Subroutines: PostVocR, PostVocL, PostVocGlide)

11. Additional vowel sequence rules are then applied in order to correct mistakes or inadequacies resulting from the first set of vowel rules.

a) If a VOWEL segment was not relabeled due to a large glitch in F1 or F2 found by "CurveType", and there are any other segmentations of the region (particularly including nasals) then the VOWEL segment is deleted. (Rules: delVOWEL, delVOWEL2)

b) For VOWELS still remaining, the formants with the glitches are smoothed, and "DoVowels" is called again. If unsuccessful after smoothing 3 times, it is assumed that the first stage of the segmentation was wrong and the segment is really a fricative. It is relabeled as such by calling LabelFRIC. (Rule: Smoothvowel)

c) If a two-vowel sequence exists, where the first label is AA or AE and the second is IY, IH, IX, EH, or EY, an attempt is made to relabel the sequence as AY by calling "CheckOut". (Rule: proposeAY)

d) EH-IYorIHorIX sequences may be optionally relabeled EY depending on the result of a call to CheckOut. (Rule: proposeEY)

e) If a BADVWL segment was created by "MakeVowelSeg" because it was rejected by "CheckOut", and there is no alternate path, it is split into two sections, under the assumption that CurveType didn't detect a vowel-vowel boundary. Each section is then processed by "DoVowels". (Rule: splitBADVWL)

f) If the lattice starts out with EL followed by a vowel, it is changed to L. (Rule: noinitialEL)

g) If the "velar" rule (9g) changed a label to a velar, but the preceding vowel was subsequently labeled as IY, AY, EY, OY, R, AXR, ER then the change is reversed, since the existence of one of those phonemes might have caused the "velar" rule to incorrectly label a nonvelar as velar. (Rule: notvelar)

h) An HHQ segment followed by a Y segment, is changed to a KG or G (depending on whether any aspiration was detected), since [HH] and glottal stops shouldn't introduce any transitions of their own. (Rule: HHQYtoG)

12. For each segment, the column (vector) of the confusion matrix corresponding to the segment label is extracted. This provides a probability ratio for every phoneme for each segment. (Rule: BLTPRB)

13. Up to this point, all unvoiced plosives have been kept in two segment sequences: a SI segment followed by an unvoiced plosive segment. These are now merged to form a single segment. (Rule: DelSI)

14. For each segment, the maximum ROP during the segment is recorded. This was intended to be used by the Lexical Retrieval component to check within-word syllable stress differences, but it has not been found to be too informative, since the use of duration in deciding on vowel labels already tends to account for most of the derivable stress information. (Rule: stress)

15. For each segment, depending on its class, subroutines may recompute a few phoneme probabilities in the vector, based on the particular acoustics of the segment. (This is called selective modification. For more details, see section B, this volume.) (Subroutine: Modify)

a) The scores on 5 nasal allophones: [M, N, NX, YM, YN] are modified using the second and third formant averages and transitions. This is done for all segments whose labels are likely to be confused with the nasal allophones. (Subroutine: ModNas)

b) The scores on 8 of the 10 unvoiced plosive allophones [P,K,KA,TV,ST,TG,JH,CH] are modified, with the affricates [JH,CH] treated like unvoiced plosives for this purpose. The acoustic features used include burst frequency, VOT, F0, frequency of aspiration, formant transitions, and minimum energy during the silence. (Subroutine: ModUVP)

c) The scores on the 8 fricative phonemes plus the score for the allophone [TCH] are modified using the acoustic features maximum ROP, frication frequency (at the energy peak), duration, and low frequency energy. ([TCH] is used as the second part of the affricate [CH] when it is segmented as two segments.) (Subroutine: ModFRIC)

d) The scores on five of the front vowel allophones: [IY,IH,EH,AE,EY] are modified using six features: F1, F2/F1, duration, maximum of F2 towards the end of the vowel, rise in F2 towards the end of the vowel, drop in F1 towards the end of the vowel. (Subroutine: ModFRONT)

e) Since the rule "bridgeFLAP" (see 9j) is usually unnecessary, all paths resulting from that rule are assigned a low path probability proportional to the likelihood that the rule will be necessary. This is done by adding a negative path probability to one and only segment in each path. (Subroutine: UnbridgeFLAP)

16. Since it is possible that the end of the utterance is incorrect as given in the segment lattice, each boundary near the end is assigned an ending probability. This is the probability that, for a path between that boundary and the end of the utterance, its segments would have been found under the assumption that they are spurious. (Subroutines: SpanL, SpanR)

Appendix 4 - Parameters for Scoring

1. Parameters for Scoring Nasals

A nasal is labeled using preceding formant transitions and the average formant values.

First the left boundary is anchored as the point of maximum drop in F1 in the region of the original boundary. Then the 30 msec region preceding this point is searched for the closest F2-F3 distance, and also for the lowest F3. If F3 has gone below 2100 Hz, indicating an [R] or [ER], then the score on [NX] (as in "sing") is decreased substantially, since in English neither [R] nor [ER] can precede [NX]. If not, the minimum F2-F3 distance is used to estimate the probability for [NX], vs [M] or [N]. Since F2 and F3 can get fairly close before an [M] or [N] preceded by [IY,EY,ER,R,AXR], we have introduced two different dictionary allophones for these occurrences of [M] and [N], named [YM] and [YN] respectively. The feature "minimum F3-F2" is used to compute three probability ratios:

a) $$\frac{P(\text{minimum}(F3-F2) \mid \text{Phoneme}=NX)}{P(\text{minimum}(F3-F2) \mid \text{Phoneme}=\text{Nasal})}$$

b) $$\frac{P(\text{minimum}(F3-F2) \mid pN \text{ or } pM)}{P(\text{minimum}(F3-F2) \mid \text{Nasal})}$$

c) $$\frac{P(\text{minimum}(F3-F2) \mid pYN \text{ or } pYM)}{P(\text{minimum}(F3-F2) \mid \text{Nasal})}$$

After this, three features are used to discriminate between ([N] or [YN]) and ([M] or [YM]). If the nasal is preceded by a vowel, the first feature is the sum of the 1-frame derivative of F2 and F3 measured 2 frames before the nasal (DF2(lbt-2)+DF3(lbt-2)). This is usually higher for [N] or [YN] than for [M] or [YM]. The second and third features are the averages of F2 and F3 respectively over the nasal. These are both usually higher for [N] or [YN] than for [M] or [YM]. These three features are used to compute two more scores:

d) $$\frac{P(DF2+DF3,\text{avg } F2,\text{avg } F3 \mid pN \text{ or } pYN)}{P(DF2+DF3,\text{avg } F2,\text{avg } F3 \mid \text{Nasal})}$$

e) $$\frac{P(DF2+DF3,\text{avg } F2,\text{avg } F3 \mid pM \text{ or } pYM)}{P(DF2+DF3,\text{avg } F2,\text{avg } F3 \mid \text{Nasal})}$$

These five probability ratios are then applied to the appropriate allophones. The modification score for each of the five nasal allophones is a product of some of the five probabilities defined above:

Allophone	Probabilities
NX	1
M	2, 4
N	2, 5
YM	3, 4
YN	3, 5

2. Parameters for Scoring Front Vowels

In order to try selective modification of vowel allophone scores, we have chosen five of the front vowel allophones [IY,IH,EH,AE,EY] to be re-scored, using six features. The final three features are computed mainly to distinguish between the diphthong [EY] and the other four vowels. Here, tmaxF1 corresponds to the point where F1 reaches its maximum. The point half way between this point and the end (rbt-1) of the vowel is named "half". The parameters are:

a) F1 measured at the center

b) F2/F1 measured at the center

c) Normalized duration - This is equal to the duration unless the vowel is the last one in the utterance, in which case it is divided by 1.5.

d) Maximum of F2 between "half" and "rbt-1"

e) "F2rise" = Difference between the averages of F2 in the interval from tmaxF1 to "half" and the interval from "half" to "rbt-1". F2 always rises for the diphthong [EY].

f) "F1drop" = Drop in average F1 between the same 2 intervals. F1 drops more for [EY] than for the other vowels.

3. Parameters for Scoring Fricatives

Since it is difficult to distinguish between the affricate [CH], and the 2-phoneme cluster [T SH] (as in "what ship"), we apply a phonological rule to the dictionary which optionally changes the phoneme [CH] into the sequence [T TCH] where [TCH] is defined as a fricative. The resulting nine fricatives are discriminated as follows. First we find the point having maximum energy (ROP) and name it "peak". Then, we measure:

a) Peak-CM75 - CM75 sampled at peak

b) Peak-ROP - ROP sampled at peak

c) Duration - a major difference between voiced and unvoiced fricatives, as well as between [SH] and [TCH].

d) Mid-LEZ - the average of parameter LEZ in the 3 points centered around "peak".

4. Parameters for Scoring Unvoiced Plosives and Affricates

In our initial use of selective modification on the unvoiced plosive and affricate allophones, we have attempted to discriminate among eight of them - [P,KA,K,TG,TV,ST,CH,JH]. First, [JH] (which is the only one that is voiced) is distinguished from the other seven using the four features:

a) average F0 over the region around the burst.

b) minimum ROP during the silence

c) frequency of aspiration (CM75 in the center of the aspiration)

d) the drop in ROP between the burst and the aspiration.

With respect to these four features, [JH] is often detected as voiced. Its minimum energy during the silence is not quite as low as for the other unvoiced plosives, and its frequency of aspiration is usually around 2500 Hz. Unlike most unvoiced plosives, its energy does not decrease between the burst and the aspiration.

The remaining seven unvoiced plosives are distinguished using the features:

a) Burst Frequency (parameter CM75 measure at the burst)

b) VOT (the measured time between the burst and the following voiced sound)

c) F3-F2 (measured 2 frames before the silence) - for those unvoiced plosives which appear to be preceded by a sonorant.

d) Energy in burst (ROP measured at the burst).

Official Distribution List

Contract N00014-75-C-0533

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314

Office of Naval Research
Information Systems Program
Code 437
Arlington, Virginia 22217

Office of Naval Research
Code 1021P
Arlington, Virginia 22217

Office of Naval Research
Branch Office, Boston
495 Summer Street
Boston, Massachusetts 02210

Office of Naval Research
Branch Office, Chicago
536 South Clark Street
Chicago, Illinois 60605

Office of Naval Research
Branch Office, Pasadena
1030 East Green Street
Pasadena, California 91106

New York Area Office
715 Broadway - 5th Floor
New York, New York 10003

Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380

Office of Naval Research
Code 455
Arlington, Virginia 22217

Office of Naval Research
Code 458
Arlington, Virginia 22217

Naval Electronics Lab. Center
Advanced Software Technology Division
Code 5200
San Diego, California 92152

Mr. E. H. Goeissner
Naval Ship Research and
Development Center
Computation and Mathematics Dept.
Bethesda, Maryland 20084

Captain Grace M. Hopper
NAICOM/MIS Planning Branch (OP-916D)
Office of Chief of Naval Operations
Washington, D.C. 20350

Mr. Kin B. Thompson
Technical Director
Information Systems Division (OP-91T)
Office of Chief of Naval Operations
Washington, D.C. 20350

Advanced Research Projects Agency
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

Commanding Officer
Naval Air Development Center
Warminster, Pennsylvania 18974

Professor Omar Wing
Dept. of Electrical Engineering
Columbia University
New York, New York 10027

Assistant Chief for Tec[illegible]y
Office of Naval Researc[illegible]
Code 200
Arlington, Virginia 22217